Evaluating NavMesh and Reinforcement Learning for Ghost AI in Pac-Man: A Pathfinding Performance Comparison

Damien Galea
Institute of Information & Communication Technology
Malta College or Arts, Science & Technology
Corradino Hill
Paola PLA 9032
{damien.galea.f32099}@mcast.edu.mt

***Abstract*—**

***Index Terms*—Reinforcement Learning, Imitation Learning, 2D, Unity, Arcade**

I. Introduction

A. Theme

B. Topic Rationale

C. Research Statement

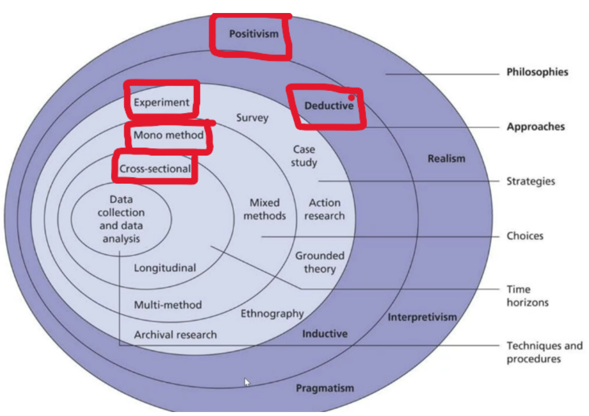

Fig. 1. Research Onion

II. Literature Review

The adoption of reinforcement learning (RL) and traditional pathfinding methods has shaped new advancements in game navigation systems. Although both techniques offer benefits, they also raise challenges around adaptability and performance in dynamic gaming environments. To better understand these developments, a targeted review of selected academic studies was carried out. This section examines the different research methodologies employed, covering both RL and traditional approaches. Fig 2 presents the Literature Map, visually summarizing the selected studies and their connections to key concepts in navigation research. This review aims to highlight common methodological trends and critically assess the strengths and weaknesses of the approaches used.

A. Traditional Pathfinding in games

The study conducted by [1] adopts an applied experimental research design with a positivist philosophy, as defined by Creswell in [2], who states that the world can be objectively measured and quantified. This research aims to test the performance of the A* pathfinding algorithm within a educational game. The methodology follows the Digital Game-Based Learning – Instructional Design (DGBL-ID) model, which provides a structured process including development, quality assurance, and evaluation stages. In this study we can see that the authors a clear procedural approach, including flowcharts, algorithmic logic and implementation code. Additionally the authors also apply scenario-based testing across three controlled game environments featuring different obstacle configurations, measuring the travel time of the NPC, which reflects a quantitative study. However the methodology in [1] lacks in some areas, such as comparing the A* algorithm to other alternatives such as Dijkstra to determine which algorithm proves more efficient in the game context, which limits an objective assessment to determine the better algorithm . Another limitation is the absence of statistical analysis, while the travel times are measured and reported no statistical tests, such as the calculation of averages, standard deviation, or significance testing, are applied to validate the results. This makes it difficult to determine the consistency and reliability of the reported results. In addition although the game is intended to be used in an educational setting the overall user experience was evaluated through the In-Game GEQ questionnaire, the study does not investigate how the performance of the pathfinding algorithm influences the educational effectiveness of the game. Some of the issues such as an NPC failing to jump is noted but not examined in relation to the player experience in regards to education and immersion.

In the study conducted by Fauzi et al in [3] uses an experimental research design with a positivist philosophy the same as study [1]. The research aim of this study is to compare the performance of A* and Basic Theta* algorithms within the

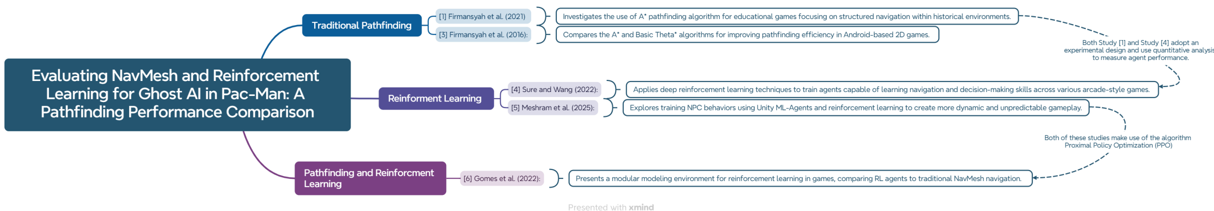

Fig. 2. Literature Map

context of a mobile game, specifically an android based 2D game. Study [3] involves implementing both algorithms inside the application and testing them across 6 scenarios designed to evaluate different obstacle conditions and map complexities. In this study 3 quantitative metrics where used: the length of the path, the number of nodes visited and the time to complete the path. These metrics where used for both algorithms to make a fair and consistent comparison solidifying the idea of a positivist philosophy. This also strengthened clarity and reliability of the data collection process. Furthermore, the decision to run the simulations multiple times contributed to the dataset although there was no formal statistical analysis. As previously mentioned there was no formal statistical analysis such as averages, standard deviation or significance testing to validate the results causing the study to have a limited conclusion similar to that of study [1]. Without statistical validation, it is difficult to determine whether the observed differences between the A* and Basic Theta* algorithms are statistically meaningful or merely due to random variation. Additionally the maps that were used in the simulation all static without dynamic obstacles reduces the study validity for complex environments. While this approach is reasonable given that the study is intended to model simpler 2D pathfinding games such as Pac-Man or Maze games it still limits the generalizability of the findings particularly when considering the requirements of modern dynamic games where adaptability is a critical factor for AI pathfinding algorithms.

As seen from both these studies they both are experimental studies with a philosophy of a positivist. Both studies are mostly lacking in the same areas were they do not preform formal statistical analysis meaning there conclusion are limited to a certain extent. It is important to note that both of these studies had a clear approach study [1] showed code snippets and flowcharts and study [3] gave clear quantitative measures.

B. Reinforcement Learning in games

In [4] a study on Reinforcment learning for general game agents was conducted with a positivist philosophy while being an experimental type of study. The methodology of this study integrate General Video Game AI (GVGAI) framework which provides a set of arcade games designed to test an agent general capabilities. This particular framework was chosen due to it ability to evaluate an agent across multiple games, forcing agents to learn strategies directly by environmental interaction rather than learn through human interaction. The authors in this study use Proximal Policy Optimisation (PPO) algorithm to train the agents widely recognized reinforcement learning method known for its stability and efficiency. The agents were trained separately on three games Aliens, DeceptiZelda and Flower with hyperparameters tuned for each game to get the best result possible. Despite its strengths, the methodology in [4] presents several limitations such as the training durations ranging from 300,000 to 1 million timesteps. This is a limitation because some agent might require significantly longer periods to develop stable and generalized policies. Moreover, the study does not conduct formal statistical validation across multiple runs or random seeds, which raises concerns about the consistency and reliability of the results. Without statistical testing, it remains unclear whether performance improvements are meaningful or merely due to random variation. Additionally, the study focuses on only three games, potentially limiting the generalizability of the findings to broader, more diverse gaming environments.

The methodology of study [5] reflects an experimental design with a positivist philosophy, it is a quantitative type of research due to ins numerical data collection this includes reward scores, episode length, policy loss and the agents performance in a controlled training environment. The ml-agents toolkit is used to implement the RL in the game, providing an environment that helps the agent to be trained through interaction. It is also specified that the algorithm PPO was used to help train the agent. The training environment consists of a simple enclosed Unity scene where agents learn to navigate and collect pellets, replicating tasks similar to hide-and-seek gameplay. To increase training efficiency the environment is duplicated across 60–80 instances enabling parallel learning a methodological strength that accelerates convergence and supports large-scale experimentation. The performance was measured by important RL metrics such as episode length and cumulative reward these provide evidence of the agent learning progression. It is important to not that in this methodology we can see that agents are not confined to fixed rules but learn behaviour dynamically through continuous feedback, addressing major limitations found in traditional, scripted NPCs thus it can be said that this methodology can be applied to learn various behaviours. Although the methodology effectively demonstrates basic reinforcement learning concepts, it suffers from several limitations. The training environment is overly simplistic, consisting only of a four-walled area and pellet collection, which does not reflect the complexity of real-world game environments. Training the agent on a single task restricts the demonstration of versatile behaviour expected from adaptive NPCs. Additionally, the reward structure is straightforward, focusing solely on immediate success without modelling more complex goals such as strategic planning or multi-objective decision-making. Finally, the evaluation relies primarily on basic metrics, missing deeper insights into the agent's learning robustness and emergent behaviours.

Study [4] and Study [5] both employ reinforcement learning techniques, specifically Proximal Policy Optimization (PPO), to train autonomous agents; however, their methodological designs differ significantly. Study [4] applies PPO across multiple games within the GVGAI framework, exposing agents to diverse environments and tasks such as enemy avoidance, strategic reward collection, and adaptive planning. In contrast, Study [5] trains agents within a simple, static Unity environment focused solely on pellet collection. As a result, Study [4] demonstrates a broader environmental complexity and evaluates deeper learning behaviours compared to Study [5], which emphasizes basic navigation success. While both methodologies highlight the benefits of RL for agent training, Study [4] offers stronger evidence for agent adaptability and generalization across varied gaming contexts.

C. Reinforcement Learning and pathfinding for games

A study conducted by Gomes et al. [6] demonstrates a positivist philosophy like the previously discussed studies, as it analysis performance by data aligning with Creswell's [2] definition of positivism. The study adopts a quantitative, experimental design, where the behaviour of game agents trained through RL is directly compared to agents using NavMesh-based pathfinding. This experimental approach provides a controlled environment to assess the performance of both navigation agents under consistent game scenarios, it is stated that the algorithms such as PPO and Asynchronous Advantage Actor-Critic (A3C)will be used to train the agent. One key strength of the methodology is the use of a modular and flexible modeling environment (AI4U), which allows reinforcement learning agents to be trained without needing manual programming of navigation paths. Moreover, by integrating standard deep RL algorithms into widely used engines like Unity and Godot, the study increases its practical relevance to current game development practices. Additionally, the method of designing clear, task-based navigation challenges offers a structured and effective way to evaluate how well agents adapt and learn over time, following good practices in artificial intelligence research. However this methodology has certain limitations these being the environment design for training can be considered simple. This is due to its basic navigation tasks that may not represent game words that are popular today, since it has a simple design agents have limited exposure to diverse challenges such as dynamic obstacles, varying terrain complexities, adversarial agents, or multi-step strategic planning, which are commonly encountered in modern game development. Another limitation is that the navigation tasks seem relatively isolated and do not evaluate agent behaviour within full gameplay loops, such as combining navigation with combat, resource management, or decision-making under uncertainty. These factors limit the applicability of the findings to more sophisticated gaming contexts where agents must learn to balance multiple objectives simultaneously. While the simplified design offers clarity for initial demonstrations, it reduces the overall ecological validity of the study.

III. Research Methodology

After carefully reviewing methodologies in multiple studies that where outline in the Literature Review, it was observed that some researchers ([1], [3], [4], [5] , [6]) commonly employed controlled experiments involving repeated testing to ensure reliable evaluation of navigation performance. These studies demonstrated the value of structured experimental setups where agent behaviour could be assessed objectively under predefined conditions. Following a similar approach, the present study designs a controlled testing environment specifically aimed to address our research questions, which are stated to be:

1) RQ1 (Navigation Time Efficiency): How does the time taken to catch a moving player differ between agents using NavMesh-based pathfinding and agents trained via reinforcement learning?
2) RQ2 (Path Efficiency): How does the distance travelled to reach the player compare between agents using traditional pathfinding methods and reinforcement learning?

To address these research questions a single experiment will be conducted to measure the time taken and distance travelled at the same time for both the Navmesh and RL agents. The experiment will be repeated a number of times across simple and complex environments to ensure reliable and consistent results. By collecting both performance indicators in a unified setup and applying repeated testing, the study provides a controlled and quantitative basis for comparing navigation strategies.

A. Methodological approach

The approach take in this research study is based on positivist principles, focusing on measurable outcomes and systematic experimentation. A deductive reasoning process was used, first we researched existing theories on agent navigation and RL than we designed our own experiment to test these theories through quantitative data gathered from a controlled environment. This study's methodology is built according to the Research onion shown in Fig 1 ensuring coherence from philosophical foundation to the data gathering techniques , thereby supporting a structured framework that objectively evaluates agent behaviour across the two different navigation approaches.

An experimental methodology was selected because it allows use to test our hypotheses through the collection of objective and quantifiable data, specifically measuring navigation time and distance travelled by agents under controlled conditions. Other research studies where deemed unsuitable for the purpose this study. Surveys would have introduced subjective biases and not offered direct measurements of agent performance. Similarly, case studies, while beneficial for in-depth examination of individual instances, are less effective for comparison research that requires statistical consistency across multiple trials. Therefore, experimental methodology offers the best solution to address our research questions and ensuring that the performance differences between Navmesh and RL agents are assessed with clarity and objectivity.

B. Experimental Design

This research project was conducted using Unity Engine as the game platform where the simulation will take place. This setup was designed to allow a direct comparison between a Navmesh agent using Unity's built in Navmesh system and a RL agent trained with the Proximal Policy Optimization (PPO) algorithm. Both agents were evaluated in a pursuit task requiring them to catch a moving target within a bounded environment.

Unity was selected due it support for game development, its build in Navmesh navigation system and it compatibility with AI and machine learning frameworks through the ML-Agents toolkit. The RL agents was conducted using a Python 3.10 environment, which was managed by an application called Anaconda ensuring compatibility with the needed machine learning libraries and stable training sessions. Training scripts for the RL agent were executed within this Python environment, making use of the ML-Agents Python API along with standard libraries such as TensorFlow and NumPy. The PPO algorithm was chosen because it is known for providing good training stability and sample efficiency, which is important for continuous control problems like dynamic navigation tasks.

In this experiment 2 types of environments where created to test the agents performance under 2 levels of difficulty. The simple environment was a small-sized map with only 1 type of obstacle, offering a basic but important challenge. This setup allowed for the testing of fundamental navigation skills in a more controlled and predictable setting, while still keeping the gameplay context similar to Pacman, where movement decisions are frequent but relatively simple. On the other hand, the complex environment included 3 different types of obstacles placed across the map. These obstacles forced the agents to make smarter decisions, like finding the best path around barriers and adjusting their movement based on the target's position. Using both environment types was important to check if the agents could perform well not just in easy conditions but also in more realistic and challenging setups. Using both types of environments was an important part of the experimental design because it allowed the study to assess not only how well agents could perform in straightforward conditions but also how they would adapt when facing challenging layouts.

The trials initiated when the 2 agents where set in the environment, the performance measurement where handled automatically by a unity scripts to ensure consistent data gathering. No manual inputs were made during the run to keep the test authentic as possible, it import to note that all the decisions made by the agents where all due to there own system either the Navmesh or the reinforcement learning policy. The experiment was ran 10 times for the simple environment and 10 times for the complex environment, resulting in 20 trials per agent while each run recorded data in real time by make use of a custom script. The script recorded the time taken to catch the moving target, the total distance travelled, the environment type (simple or complex), and the run number. Each run had a different file to prevent data overwriting each other and made post-experiment data management easier.

It is important to note that the trials are not terminate once 1 agent catch the target. Instead both the Navmesh and RL agents were allowed their pursuit independently, either by catching the target themselves. This ensured that performance data for both navigation methods could be collected fairly and without interruption.

C. Method of analysis

After all the run are completed, the data collected will be organized to later be placing in the software IBM SPSS Statistics. This software will be used to process the data due its reliability and support for handling quantitative dataset. For each agent and each environment (simple and complex), the mean values for time and distance will be calculated across 10 runs. The standard deviation will also be computed to examine consistency in agent performance. Comparison between the agents will be mad within each environment to highlight how the agent efficiency changes under different levels of difficulty. Bar charts generated in SPSS will be used to visually represent the differences in average time and distance travelled, providing an accessible view of the performance gaps. This method of analysis directly addresses the research questions by objectively quantifying efficiency differences between navigation approaches. Using statistical summaries and visualizations strengthens the conclusions by supporting them with measurable evidence, fully aligning with the positivist foundations of the study.

D. Validity, Reliability, and Generalizability

1) Validity: To insure that validity of this experiment many steps where taken. This include spawning the Navmesh and RL agents start off from the same position. Furthermore, the experimental setup required both agents to complete the task independently by successfully capturing the moving target before the trial ended, preventing early termination biases. Additionally, custom Unity scripts were used to automatically record time taken and distance travelled for each agent, ensuring consistent and objective data capture. Finaly the data collected was store across multiple files as stated earlier, allowing manual verification of results after trials.

2) Reliability: The reliability of this experiment was strengthen through repetition, controlled conditions, and consistent data collection methods. Due to the chance that small changes could happen because of the movement target, the agent was testing 10 times for each environment. Repeating the trials allowed patterns to be seen clearly and reduced the chance that results were just random. It is important to note that the setup for each run was identical with the exception to when we change to the 2nd environment. All performance results, like time taken and distance travelled, were recorded automatically using Unity scripts to avoid mistakes from manual recording. Although the target was moving randomly around the level of difficult stayed the same.

3) Generalizability: When considering the generalizability and transferability of the findings, some limitation must be considered. The environments used in this study were based on simple 2D designs similar to classic arcade games like Pac-Man. An important note to add the results will be applicable to small 2D structured navigation tasks. The experiment also focused only on basic pursuit behaviour and did not test agents on broader gameplay tasks like resource collection or combat. Therefore will this study might be valuable to others it is important to keep in mind that caution is needed when apply this to more of dynamic environments. Future research could improve generalizability by including larger, more varied environments and multi-task challenges.

E. Ethics

This research does not present any ethical concerns as it does not involve any human participants, sensitive data or any world risks. All experiments that were conducted were simulated within a game using virtual agents. Additionally, no copyrighted or third-party content was used in the simulation, and all software tools (such as Unity ML-Agents and Anaconda) were used in compliance with their licensing agreements.

IV. Findings & Discussion of Results

V. Conclusion

References

[1] D. Kurniadi, A. Mulyani, and R. S. Maolani, "Implementation of Pathfinding Algorithm in Sundanese Land History Educational Game," in *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, Sep. 2021, pp. 145–150. [Online]. Available: https://ieeexplore.ieee.org/document/9590181

[2] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Oct. 2022, google-Books-ID: Pr2VEAAAQBAJ.

[3] E. R. Firmansyah, S. U. Masruroh, and F. Fahrianto, "Comparative Analysis of A* and Basic Theta* Algorithm in Android-Based Pathfinding Games," in *2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, Nov. 2016, pp. 275–280. [Online]. Available: https://ieeexplore.ieee.org/document/7814916

[4] E. K. Sure and X. Wang, "A Deep Reinforcement Learning Agent for General Video Game AI Framework Games," in *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Jun. 2022, pp. 182–186. [Online]. Available: https://ieeexplore.ieee.org/document/9844524

[5] R. Meshram, S. Krishna, O. Kulkarni, R. R. Patil, G. Kaur, and S. Maheshwari, "NPC Behavior in Games Using Unity ML-Agents: A Reinforcement Learning Approach," in *2025 International Conference on Automation and Computation (AUTOCOM)*, Mar. 2025, pp. 1519–1523. [Online]. Available: https://ieeexplore.ieee.org/document/10956320

[6] G. Gomes, C. A. Vidal, J. B. Cavalcante-Neto, and Y. L. B. Nogueira, "A modeling environment for reinforcement learning in games," *Entertainment Computing*, vol. 43, p. 100516, Aug. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952122000404