Realization and Evaluation of a Finite State Machine as a player for a Tactical Game

Lars Büsselberg[1], Pascal Reuss[1]

[1]*Institute of Computer Science, Intelligent Information Systems Lab, University of Hildesheim, Hildesheim, Germany*

Abstract

The integration of artificial intelligence (AI) methods into entertainment software, especially gaming scenarios, presents new dimensions for dynamic and immersive experiences. This paper explores the development and evaluation of an AI-based tactical scenario within the Unreal Engine platform, focusing on functional and non-functional requirements to assess the suitability of the implemented AI. Through a case study involving competition scenarios and unit behavior tests, the AI demonstrates capacity for intelligent decision-making, complex strategies, and adaptability. However, limitations in learnability and team coordination are identified, suggesting potential enhancements through advanced AI techniques such as reinforcement learning and/or case-based reasoning. Overall, this study gives a foundation for future developments in AI, emphasizing the importance of continual research and optimization to enhance the possible outcome of AI implementations.

Keywords

AI-based decision making, Tactical game, Turn-based game, Knowledge Modeling, Software agents

1. Introduction

The progressive integration of artificial intelligence methods into various application domains is shaping the direction of intelligent information systems development. Particularly, the realm of entertainment software, especially computer or video games, benefits from AI utilization by opening new dimensions through exciting, challenging, dynamic, and realistic gaming scenarios. This development is not only propelled by the ongoing technological advancements but also by the increasing expectations of players to receive ever-better, more interactive and immersive gaming experiences. However, this trend can also be viewed from the perspective of AI research: demanding scenarios pose a significant challenge for the development of suitable intelligent and adaptive software agents. For this purpose, either established solutions can be applied to the given scenario, or an independent approach can be developed using known AI methods. This interaction, consisting of increasingly sophisticated gaming environments that require and enable ever more advanced AI agents, provides an ideal breeding ground for AI research. The results obtained from this can not only be used in the context of computer games but can also be transferred across domains and investigated through independent disciplines of artificial intelligence.

This paper presents a turn-based tactical scenario designed for the application of AI methods. A rudimentary multi-agent AI based on a finite state machine is implemented as a base or test component. The aim is to define the functional and non-functional requirements for an AI to then evaluate the scenario accordingly. This holistic approach aims to determine whether the prototype scenario is sufficiently complex for the application of various AI methods and what measures could be taken if a deficiency is uncovered. Furthermore, conclusions are drawn from this implementation regarding whether additional methods of artificial intelligence could be applied and what additional synergies could result from them.

2. Related Work

This overview of related work on artificial intelligence in video games is based on comprehensive research conducted through the systematic application of specific search terms in well-known scientific

LWDA'24: Lernen, Wissen, Daten, Analysen September 23–25, 2024, Würzburg, Germany
buessel@uni-hildesheim.de (L. Büsselberg); reusspa@uni-hildesheim.de (P. Reuss)

databases, including Google Scholar, Semantic Scholar, and Science Research. The chosen search terms were selected to provide a broad overview of current research and developments in the field of AI in video games and to contextualize the results of this work. These terms include key phrases such as "Artificial Intelligence in Video Games," "Game AI Techniques," "Behavior-Driven AI in Gaming," "Evaluate Games with AI," and "Scientific Research using Unreal Engine."

In their work, Yannakakis & Togelius [1] provide an overview of the main application areas of AI in video games, focusing on enhancing player experience, procedural content generation, and using machine learning for generative artifact creation. Perez-Liebana et al. [2] implemented "Tribes," a turn-based strategy game, to investigate AI applications, presenting challenges for artificial intelligence such as unit coordination, long-term planning, opponent modeling, and managing extensive action spaces. This work suggests potential extensions, including the implementation of model-free reinforcement learning (RL) or deep RL agents, given the framework's suitability for decision-making algorithms.

Following the patterns outlined by Yannakakis & Togelius, several works presented in the subsequent sections align with their classifications:

Forgette & Katchabaw [3] introduce a framework for evolving believable agents using RL, achieving promising results in creating human-like non-player character (NPC) agents. Yandong and his colleagues [4] propose a hierarchical learning approach combining imitation learning and RL for designing credible agents in team sports games. Bakkes et al. [5] discuss challenges in creating effective and intelligent behavior in NPCs, advocating for a case-based AI approach to adapt to changing game circumstances. Egenfeldt-Nielsen ans his colleagues [6] describe various AI development methods in video games, including finite state machines, rule-based systems, and chance-based systems. Ontanon [7] demonstrates the use of natural language processing (NLP) for creating narrative-driven, human-like NPCs in games.

Additionally, approaches such as evolutionary AI algorithms [8] [9], emotional AI [10], script-driven AI [11], and search-based AI [12] are explored for their applications in video games.

Furthermore, the use of the Unreal Engine extends beyond game development into scientific research, as evidenced by studies such as those by Boyd & Barbosa [13] and Bondi et al. [14]. These works demonstrate the engine's versatility in providing simulation environments for testing AI algorithms and supporting the implementation of AI techniques through tools like Blueprints and Behavior Trees.

3. Presentation of the Tactical Scenario

The presentation of the tactical scenario encompasses an in-depth exploration of the game's strategic landscape, highlighting its conceptual framework, technical integration, and gameplay mechanics. This section delves into the fundamental aspects of the scenario, including the initiation of AI flows, spell selection processes, movement algorithms, and decision-making logic for unit actions. The scenario was implemented using the Unreal Engine, leveraging its capabilities to realize the envisioned gameplay experience. By elucidating these core elements, the tactical scenario's design, implementation, and operational dynamics are elucidated, providing a comprehensive understanding of its underlying structure and functionality.

3.1. Overview and General Concepts

The game presents a turn-based tactical scenario accommodating 2 to 4 teams, known for their challenging and strategic gameplay dynamics, making them a fitting choice for this application. Within the game environment, rectangular fields with grid projections form the playable terrain, defining the movement possibilities for units. Units, categorized by type, can navigate these fields either orthogonally or diagonally, depending on their specific abilities. Terrain features such as obstacles, including rocks and trees, add strategic depth by impeding movement and obstructing attacks. The implementation of Line of Sight mechanics ensures that units must have direct visibility to engage in combat, preventing attacks through obstructive elements. In the combat phase, teams take sequential turns, with each team's number of actions matching their unit count. Units expend Move Points (MP) to traverse the terrain, strategically positioning themselves for optimal engagement. Additionally, they utilize Action Points (AP) to cast spells, each with varying AP costs and range limitations. Unit types, namely Warriors, Archers, and Slimes, each possess unique characteristics and abilities, contributing to the game's strategic diversity. Warriors excel in close combat with high Hitpoints (HP) but limited mobility, while Archers specialize in ranged attacks, albeit with lower HP. Slimes offer a balance between offense and defense, featuring moderate mobility and versatile spells. Spellcasting plays a pivotal role, with spells differing in their effects, AP costs, and range, fostering tactical decision-making and strategic depth. Some spells have area of effect properties, affecting multiple units within a designated range, further emphasizing the importance of positioning and spatial awareness. Overall, the game's mechanics and dynamics aim to provide players with engaging and strategic gameplay experiences, encouraging thoughtful decision-making and tactical prowess.

In addition to the general concept, the Pathfinding blueprint implements a planning-based A* search algorithm [15], utilized for unit movement and position examination on the game board. It ensures optimal paths for both player-controlled and AI-controlled units, minimizing wasted movement points. The algorithm follows a step-by-step logic:

1. **Initialization**: Start and target fields are set, and two lists are created: to-be-explored and explored fields.
2. **Setting start field as the current field**: It is added to the to-be-explored list, with actual costs set to 0 and estimated costs calculated using Euclidean distance.
3. **Loop until the list of to-be-explored fields is empty**:
   a) Choose the field with the lowest total cost (sum of actual and estimated costs) and set it as the current field.
   b) If the current field is the target, the path is found; proceed to step 4.
   c) Otherwise, move the current cell from the to-be-explored list to the explored list and examine neighbors.
   d) For each direct neighbor (including diagonal neighbors if applicable):
      i. Ignore it if already explored or impassable.
      ii. Calculate preliminary actual costs from the start point to this neighbor.
      iii. If the neighbor is not in the list of to-be-explored fields, add it and set the actual costs.
      iv. If the neighbor is already in the list of to-be-explored fields and the new actual costs are lower, update the actual costs.
4. **Backtracking**: Construct the path from the target field back to the start field by tracing predecessor fields.

The algorithm assumes primarily two-dimensional fields, employing Euclidean distance as the heuristic for calculating costs between two fields. While the pathfinding algorithm is fundamental in AI, in this paper, it is discussed as a means to an end rather than explored in depth. Notably, the A* algorithm, with its optimized search space expansion towards the target, is preferred over the Dijkstra algorithm for large or obstacle-laden game boards due to its efficiency.

3.2. AI Integration

In integrating AI into the Unreal Engine, we initially considered using existing libraries and methods such as BehaviourTrees or StateTrees. However, due to their complexity, we opted to develop a simpler AI system based on finite state machines using Blueprints for our units. Finite state machines are commonly used for AI [16], offering a structured approach to decision-making. Our AI framework follows a predetermined flow for each unit, progressing through defined states to execute actions such as choosing spells, moving, and casting spells. Each state is triggered by specific conditions, ensuring a logical sequence of actions. Our AI flow consists of the following states:

1. **InitiateAIFlow**: This marks the entry point of the AI. No direct logic is executed here; rather, it simply initiates the processing of the AI flow. From here, the computer-controlled unit can transition to the next state.

2. **ChooseSpell**: Immediately after the start of its own turn, the AI checks which spell it should cast and where on the field. For this, an Impact Score is calculated, composed of various factors describing the effects of casting a spell on a specific field. The spell and location with the highest Impact Score are chosen. If it's not possible to cast a spell because no opponent or ally can be reached due to insufficient movement points (thus resulting in a negative or 0 Impact Score), the process jumps directly to state 5.
3. **MoveToSpellPosition**: In this step, the pathfinding algorithm is used to reach the previously calculated best field, and the unit is moved there.
4. **CastSelectedSpell**: This state handles the casting of the selected spell on the previously calculated field. If there are still action points available and another spell can be cast, the process jumps back to state 2 and repeats. If all action points are used up, the process moves to the next state.
5. **DetermineEndPosition**: Depending on the logic of the unit type, in this state, the decision is made on where the unit should end its turn. For example, it might be advantageous for a warrior to stay close to the opponent to continue close combat attacks. For archers and slimes, it might be beneficial to create as much distance as possible from the surrounding enemies, provided there are still movement points available at the end of the first movement phase. The pathfinding algorithm is used again to determine the best field for this purpose.
6. **MoveToEndPosition**: The AI moves the unit to the optimal position calculated in the previous step to complete the turn.
7. **EndTurn**: Similar to state 1, this state is only for ending the AI flow. No direct logic of the AI unit is executed here; rather, the turn is simply ended, allowing the next player, AI opponent, or unit to take over.

Our AI system is designed to be unit-specific, allowing individual units to be controlled by AI or human players. This flexibility enables diverse gameplay scenarios, including AI vs. AI matches. To enhance decision-making, we introduced an impact score calculation in state 5, considering factors like HP damage and the potential to eliminate enemy units. By assigning adjustable coefficients to these factors, we can fine-tune the AI behavior to prioritize certain actions over others. In summary, our integration of AI into the Unreal Engine provides a structured approach to unit behavior, allowing for dynamic and engaging gameplay experiences. By leveraging finite state machines and customizable decision-making criteria, we aim to create challenging yet balanced encounters for players.

One challenge in AI development was distinguishing between various unit types regarding AI logic. For instance, a warrior's behavior differs from that of an archer in terms of movement, positioning, and attack strategies. Developing a comprehensive AI flow to encompass the unique characteristics of each unit type is likely to be complex and error-prone. The solution involves creating a base AI blueprint class outlining the fundamental AI flow, with subsequent unit types inheriting from this class to implement specific logic deviations. However, it's essential to specify the AI class for each unit type during unit creation to avoid defaulting to the base class.

Another aspect of AI integration, identified as "Level generation by AI," presents an intriguing topic within the realm of AI integration. However, it's excluded from this study for two primary reasons. Firstly, the study's scope limitations necessitate a focused analysis to ensure thoroughness within the allotted time frame. Secondly, excluding level generation allows for a more concentrated examination of the other identified application areas, addressing central questions and challenges in AI integration. Nonetheless, level generation by AI remains a potential area for future research or in-depth studies, given its significant role in AI applications.

4. Evaluation

4.1. Experimental Setup

The experimental setup is paramount for evaluating the AI application within the context of the turn-based tactical gaming scenario introduced in Section 3.1. Fundamental decisions are made to ensure

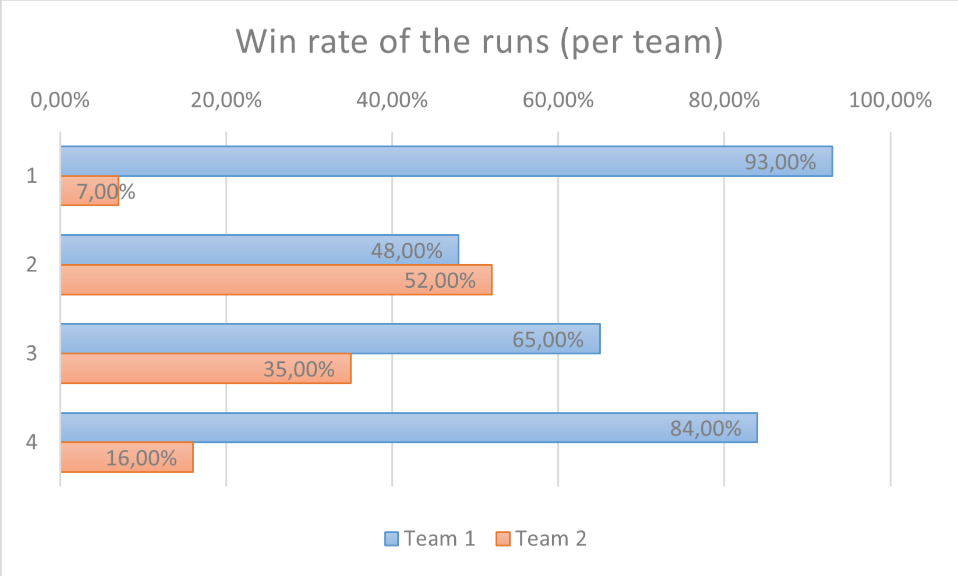

Figure 1: Win rate of the runs in an non-alternating turn sequence with changing starting teams (2 team configuration).

the integrity and robustness of the experiments. This involves defining the composition, coordination mechanisms, and decision-making processes of AI teams to simulate realistic gameplay scenarios. The design of gaming environments encompasses diverse terrain features, strategic chokepoints, and tactical advantages to effectively challenge the AI. Standardized starting conditions and unit attributes are established to create equitable gameplay, facilitating fair comparisons and accurate performance assessments. Multiple experimental variants are devised to explore different aspects of the AI's capabilities, including scenarios with varying team numbers, turn sequences, and environmental conditions. A systematic data collection framework captures both quantitative metrics (such as win rates, resource utilization, and turn efficiency) and qualitative observations of AI behavior and decision-making processes. Overall, meticulous attention to detail in the experimental setup ensures the validity, reliability, and reproducibility of the results obtained.

In addition, the evaluation of AI integration is outlined based on both quantitative and qualitative success criteria. Quantitative criteria include the win-loss ratio, categorized based on the ratio of wins to losses over a defined number of rounds. These criteria provide insights into the suitability and performance of the AI for the game. Furthermore, cross-game metrics such as resource management and performance consistency, as well as game-specific metrics such as measurable values per unit and prioritization behavior, are discussed.

Qualitative success criteria include individual behavior tests and scenario tests. Individual behavior tests allow for examining the AI's behavior in specific situations and comparing it to expected behavior, while scenario tests evaluate the AI's response to specific game situations and analyze its effectiveness in making intelligent tactical decisions. These qualitative criteria provide comprehensive insights into the AI's performance and complement the quantitative evaluation methods.

4.2. Competition scenario with varying team numbers and turn sequence

The assessment of AI performance in the first configuration of the first scenario, involving two teams, provides a thorough insight into the AI's behavioral patterns and decision-making mechanisms. Through meticulous analysis of win rates across diverse game variations and examination of quantitative metrics such as damage dealt, healing, and strategic adaptations, several crucial observations emerge. The primary observation is the AI's adaptability and strategic acumen in response to varying gameplay conditions. The variance in win rates across different scenarios highlights the AI's ability to dynamically adjust its strategies to match the evolving game state. This adaptability is essential, especially in scenarios

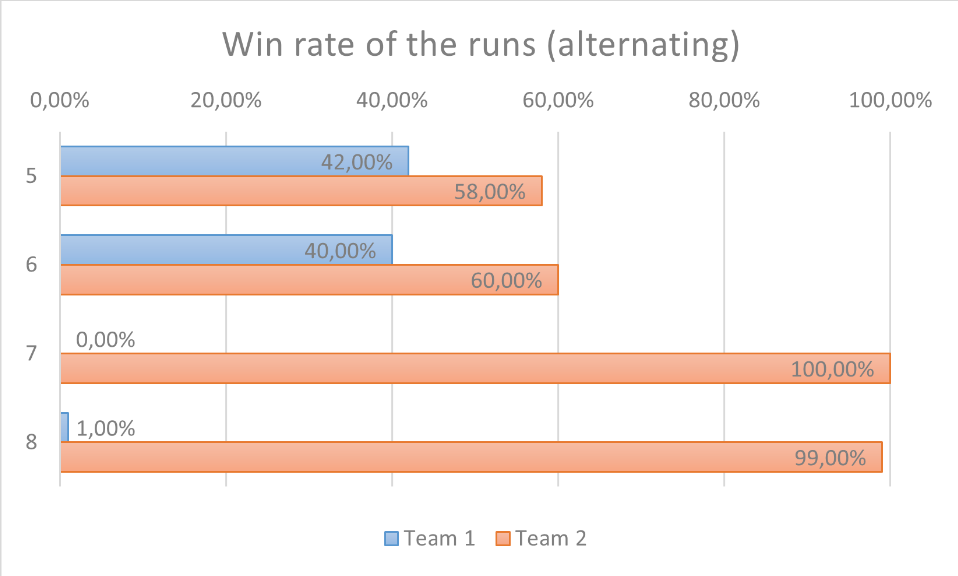

Figure 2: Win rate of the runs in an alternating turn sequence without changing starting teams (2 team configuration).

where factors like terrain or team composition change, showcasing the AI's flexibility and tactical awareness.

Moreover, a closer examination of quantitative metrics reveals insights into the AI's decision-making processes. Comparing win rates (see Fig. 1 & Fig. 2 with accumulated team damage (see Fig. 3) reveals a trend: higher total damage correlates with higher win probabilities in several runs. This aligns with the functional requirement of adapting strategies based on contextual factors. Variations in damage values within and across runs demonstrate the AI's dynamic adaptation to gameplay conditions. While win rates often correlate with the total damage inflicted by AI-controlled units, it becomes evident that factors like strategic healing and tactical adjustments also play significant roles in determining game outcomes. Instances where the AI prioritizes defensive maneuvers or exploits opponents' weaknesses illustrate its strategic depth and ability to optimize resources for maximum impact. Furthermore, the evaluation of non-functional requirements, particularly the AI's learnability, sheds light on its potential for continuous improvement. Although the current iteration of the AI prioritizes user-friendliness over advanced learning capabilities, future iterations could leverage learning mechanisms effectively to introduce varying difficulty levels, enhancing realism and scalability without compromising accessibility.

Moving on to the assessment of AI performance in the second configuration of the first scenario, where four AI-controlled teams compete, provides a detailed understanding of the AI's strategic adaptability and decision-making capabilities. Through a comprehensive analysis of win rates across multiple iterations and thorough examination of quantitative metrics such as damage dealt, healing, and strategic adjustments, several significant insights into the AI's behavior and performance emerge.

In the first analysis regarding overall win rates (see Fig. 4), Team 2 and 4 are tied with 19% of wins each, followed by Team 1 with 28.63%. Team 3 has won the most games with 33.38%. Overall, the values are around the expected value of 25% per team. However, to make further statements about this, the individual runs needed to be examined more closely regarding the quantitative success factors.

One notable observation is the AI's remarkable adaptability to diverse gameplay conditions and its strategic prowess in dynamic scenarios. The variability in win rates across different iterations underscores the AI's ability to dynamically adjust its tactics to suit the evolving dynamics of the game, showcasing its strategic depth and versatility.

Moreover, a detailed analysis of quantitative metrics provides valuable insights into the AI's decision-making processes and tactical acumen. Instances where the AI strategically prioritizes defensive maneuvers or exploits opponents' weaknesses illustrate its ability to prioritize tactical advantage over brute force, demonstrating its strategic depth and adaptability in complex gaming environments. Addi-

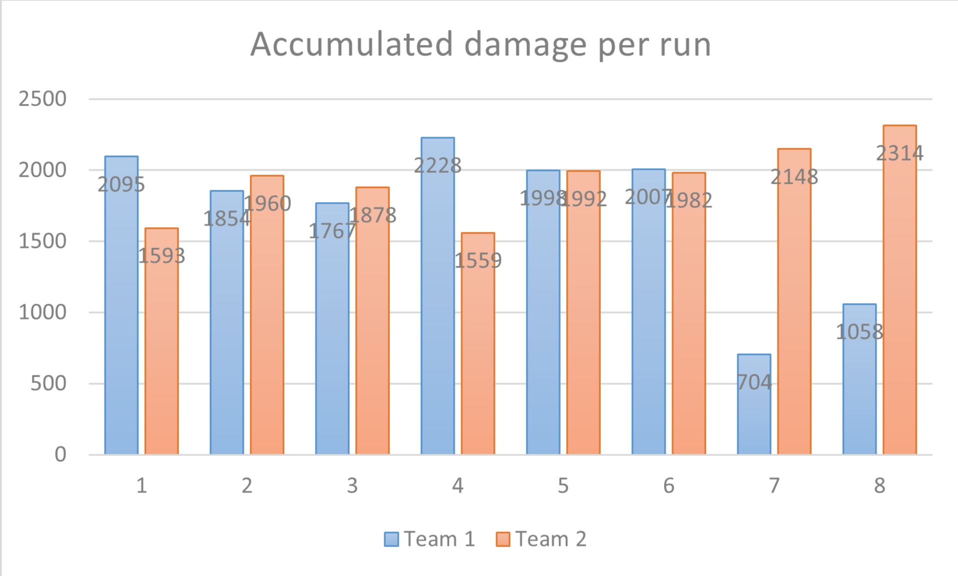

Figure 3: Accumulated damage of all units per team in all 8 runs (2 team configuration).

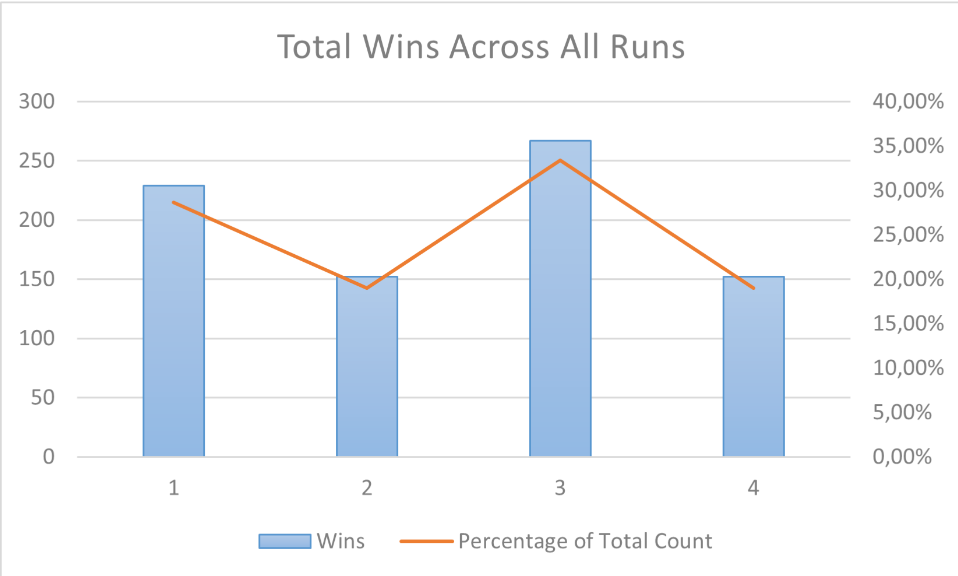

Figure 4: Total wins of all runs (4 team configuration).

tionally, the assessment of non-functional requirements, particularly concerning the AI's learnability, offers valuable insights into its potential for continuous improvement. While the current iteration of the AI may prioritize user-friendliness over advanced learning capabilities, future iterations could leverage learning mechanisms effectively to introduce varying difficulty levels, enhancing realism, scalability, and overall gaming experience without compromising accessibility.

In summary, the evaluation of AI performance in the second scenario provides a nuanced understanding of its strategic capabilities, adaptability, and potential for growth. Through adaptive tactics, strategic decision-making, and a balanced approach to user-friendliness and advanced learning capabilities, the AI demonstrates its effectiveness in navigating complex gaming environments and delivering engaging gameplay experiences.

4.3. Unit type behavior tests

Scenario 2 of the case study involved a comprehensive examination of artificial intelligence (AI) through individual behavior tests conducted on different unit types within the game environment. These tests aimed to delve into the qualitative aspects of AI behavior, focusing on key parameters such as movement, attack strategies, defense mechanisms, and the utilization of environmental features. By defining expected behaviors for each unit type and comparing them against observed behaviors, researchers sought to gain insights into AI decision-making processes and effectiveness.

The assessment of the Warrior unit type highlighted both strengths and areas for improvement. While the Warrior generally performed well in terms of movement, demonstrating expected behaviors in most situations, there were notable exceptions, particularly in retreat scenarios when facing overwhelming enemy numbers. To address this, the proposal of implementing a "danger value" system emerged, which could dynamically assess threats and prompt adaptive responses from the AI, potentially enhancing its strategic capabilities.

In terms of attack behavior, the Warrior largely met expectations, yet there were opportunities for refinement, especially concerning target prioritization. Introducing adjustments to the impact value calculation could enable more strategic decision-making, ensuring that the Warrior focuses its attacks on high-priority targets more effectively.

Similarly, while the Warrior's defense behavior was generally adequate, there was room for improvement, particularly in its interactions with support units like the Slime. By leveraging the proposed danger value system, the AI could make more informed decisions regarding defensive maneuvers, such as moving towards the Slime for healing when necessary, thereby enhancing its survivability and tactical effectiveness.

The evaluation of the Archer unit type revealed a more nuanced picture. While the Archer demonstrated proficient movement and attack behaviors, showcasing an ability to maintain distance from threats and target enemies effectively, shortcomings were identified in its defense strategies and team positioning. By incorporating the proposed danger value mechanism, the AI could better prioritize defensive actions and optimize its positioning relative to allied units, thereby fostering stronger synergy and tactical coordination within the team.

The assessment of the Slime unit type highlighted its overall effectiveness in executing attack strategies, leveraging its area-of-effect abilities to maximum advantage. However, opportunities for improvement were identified in its support role, particularly in prioritizing healing actions towards allied units in critical condition. By integrating the danger value system, the AI could enhance its decision-making process, ensuring timely support and maximizing team survivability.

Overall, the introduction of the danger value system emerged as a promising solution to address the identified shortcomings across unit types, offering a framework for more adaptive and strategic AI behavior. By dynamically assessing threats and adjusting decision-making processes accordingly, the AI could enhance its performance in various aspects of gameplay, ultimately contributing to a more immersive and engaging gaming experience for players.

4.4. Interpretation

The evaluation of the AI within the context of the turn-based tactical scenario highlighted several key findings. Firstly, in terms of functional requirements, the AI demonstrated the capacity for intelligent decision-making and effective execution of attack strategies. However, deficiencies were noted in areas such as environmental awareness and teamwork, where the AI struggled to adapt its behavior based on the game environment and coordinate with allied units effectively.

On the non-functional side, the AI performed well in aspects like realism, reaction time, and scalability. It exhibited predictable and fair behavior based on complete information and game rules, reacting promptly to game events without undue delays. Moreover, its performance scaled effectively across different team compositions, indicating robustness in handling varying game complexities.

Nevertheless, certain non-functional requirements such as learnability and user-friendliness were not met satisfactorily. The AI lacked the ability to learn from its experiences or adapt its difficulty level based on player interactions, limiting its overall usability and appeal. Security aspects were not directly evaluated, as they were primarily governed by the underlying game engine rather than the AI system itself.

In comparison with related research, the current approach utilized a finite state machine, which provided a structured framework for AI decision-making. However, it also highlighted the potential benefits of integrating more advanced AI techniques, such as Machine Learning or Case-Based Reasoning, to further enhance the AI's performance and adaptability. These methods could enable the AI to learn from past experiences or adapt its strategies dynamically based on evolving game conditions.

In summary, while the AI demonstrated competence in certain areas, there remains significant room for improvement. Incorporating more sophisticated AI techniques could address existing limitations and enhance overall performance and usability, paving the way for a more engaging and immersive player experience.

5. Conclusion and Outlook

The aim of this work was to explore the development and evaluation of an AI-based tactical scenario as a suitable application for artificial intelligence. A turn-based tactical scenario was conceived and prototypically implemented using the Unreal Engine platform. Three types of units with different abilities and values were designed, with a particular challenge being the development of a suitable pathfinding algorithm for the game field due to limitations of the built-in navigation system. Additionally, an AI system was implemented, employing a hybrid approach based on a finite state machine and rule-based transitions to enable the use of units as non-player entities.

Functional and non-functional requirements were established to evaluate both the scenario and the AI system for their intended use. These requirements served as an evaluation framework to measure the suitability of the AI. Data necessary for evaluation were collected through a case study involving an autonomously conducted competition scenario and a unit behavior test using defined qualitative and quantitative success criteria.

The final analysis of the functional and non-functional requirements illustrated the AI's ability to make intelligent decisions, develop complex attack and defense strategies based on the current state of the game field, manage resources effectively, and adapt to different game situations, resulting in an overall realistic and challenging player experience. However, limitations in the AI development were identified, particularly regarding the AI's learnability and team coordination.

To address these challenges, various approaches from related works were identified, focusing on integrating standalone or hybrid AI methods such as reinforcement learning or case-based reasoning. These methods have the potential to improve the AI's learnability and strengthen its teamwork capabilities.

The proposed extensions offer promising prospects for future developments in AI-driven tactical games. By integrating more advanced learning mechanisms and coordinated team strategies, the prototype presented in this work can be expanded to fulfill the non-functional requirement of learnability, potentially enhancing the player experience and opening new opportunities for AI application. In addition, the game will be connected to our visualization tool VISAB [17], to get more insights and explanations on the agents behaviour.

In conclusion, the prototype developed in this work for AI application shows promise and provides a solid foundation for further development and investigation of various AI methods. However, limitations regarding learnability were identified, which could be overcome through further research and optimizations. Overall, the development of an AI-based environment is a complex process where both the designed application and the integrated AI system have a significant impact on success.

References

[1] G. N. Yannakakis, J. Togelius, Artificial Intelligence and Games, Springer, 2018. http://gameaibook.org.

[2] D. Perez-Liebana, Y.-J. Hsu, S. Emmanouilidis, B. Khaleque, R. Gaina, Tribes: A new turn-based strategy game for ai research, Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 16 (2020) 252–258. URL: https://ojs.aaai.org/index.php/AIIDE/article/view/7438. doi:10.1609/aiide.v16i1.7438.

[3] J. Forgette, M. Katchabaw, Enabling motivated believable agents with reinforcement learning, Conference Proceedings - 2014 IEEE Games, Media, Entertainment Conference, IEEE GEM 2014 (2015). doi:10.1109/GEM.2014.7048106.

[4] Y. Ji, Z. Li, Y. Sun, X. Peng, S. Levine, G. Berseth, K. Sreenath, Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot, 2022, pp. 1479–1486. doi:10.1109/IROS47612.2022.9981984.

[5] S. C. Bakkes, P. H. Spronck, H. J. Van Den Herik, Opponent modelling for case-based adaptive game ai, Entertainment Computing 1 (2009) 27–37.

[6] S. Egenfeldt-Nielsen, J. Smith, S. Tosca, Understanding Video Games: The Essential Introduction, 2013. doi:10.4324/9780203116777.

[7] S. Ontañón, J. Valls-Vargas, J. Zhu, Bridging the gap between computational narrative and natural language processing, in: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[8] M.-J. Kim, C. W. Ahn, Hybrid fighting game ai using a genetic algorithm and monte carlo tree search, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 129–130. URL: https://doi.org/10.1145/3205651.3205695. doi:10.1145/3205651.3205695.

[9] G. N. Yannakakis, J. Togelius, A panorama of artificial and computational intelligence in games, IEEE Transactions on Computational Intelligence and AI in Games 7 (2014) 317–335.

[10] K. Isbister, Better Game Characters by Design: A Psychological Approach, 2006. doi:10.1201/9780367807641.

[11] G. Robertson, I. Watson, A review of real-time strategy game ai, AI Magazine 35 (2014) 75–104. URL: https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v35i4.2478. doi:https://doi.org/10.1609/aimag.v35i4.2478. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v35i4.2478.

[12] F. Shaukat, A. Shafique, A. Islam Qadri, Comparative analysis of search algorithms in ai, 2022. doi:10.13140/RG.2.2.29282.61123.

[13] R. A. Boyd, S. E. Barbosa, Reinforcement learning for all: An implementation using unreal engine blueprint, in: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017, pp. 787–792. doi:10.1109/CSCI.2017.136.

[14] E. Bondi, D. Dey, A. Kapoor, J. Piavis, S. Shah, F. Fang, B. Dilkina, R. Hannaford, A. Iyer, L. Joppa, M. Tambe, Airsim-w: A simulation environment for wildlife conservation with uavs, in: Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, COMPASS '18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3209811.3209880. doi:10.1145/3209811.3209880.

[15] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (1968) 100–107. doi:10.1109/TSSC.1968.300136.

[16] H. Lou, 2017. URL: https://sitn.hms.harvard.edu/flash/2017/ai-video-games-toward-intelligent-game/.

[17] J.-J. Bartels, S. Viefhaus, P. Yasrebi-Soppa, P. Reuß, K.-D. Althoff, Visualizing the behaviour of cbr agents in a fps scenario, in: D. Trabold, P. Welke, N. Piatkowski (Eds.), Lernen, Wissen, Daten, Analysen. GI-Workshop-Tage "Lernen, Wissen, Daten, Analysen" (LWDA-2020), September 9-11, Online, CEUR, 2020, pp. 130–141.