Scripted AI for Overcooked

Designing and Evaluating a Scripted AI Controller for Simplified Overcooked

Anton Mihai Cosmin

Supervisors: Frans Oliehoek, Robert Loftin

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Anton Mihai Cosmin Final project course: CSE3000 Research Project
Thesis committee: Frans Oliehoek, Robert Loftin, Klaus Hildebrandt

Abstract

Overcooked, an immersive multiplayer video game centered around cooperative cooking challenges, provides the roots for this research project. The study focuses on designing and evaluating a hand-authored controller in comparison to controllers implemented using various machine learning techniques, such as Population Based Training, in the context of a simplified version of the game. The main objective is to assess the cooperation between the hand-authored controller and a human controlled agent, with a particular emphasis on the coordination and minimizing errors during runs of fixed time length. A testing method has been designed in order to more accurately evaluate the performance. Through the implementation of a specialized controller utilizing techniques such as Behavior Trees and Decision Trees, the controller was able to successfully accomplish the task of delivering soups. An analysis was made to examine the performance of the hand-authored controller in comparison to the results obtained in other research papers that use the same simplified version of the game. After the results have been obtained, a clear difference was visible. The scripted AI performed better when paired with himself than one created with population based training that was paired with himself. However, the scripted AI was not better than a coupled planning algorithm that was paired with himself, but the computations were easier and faster for the scripted AI. When paired with a human, the overall performance decreases, but increases only for one specific map.

1 Introduction

This section provides an overview of the growing interest in human-AI cooperation, highlights the significance of studying cooperation in the Overcooked game, and introduces the motivation behind the research project to design and evaluate a scripted AI controller in the simplified Overcooked environment.

In recent years, there has been a lot of interest in studying how humans and AI can work together effectively [1]. An interesting area of research is cooperative video games, where players team up with AI agents to achieve common goals. One popular game in this domain is Overcooked [2], where players work together to manage a kitchen where they have to cook and deliver meals.

The objective of this project is to design and then evaluate a hand-authored controller in comparison to controllers trained using population-based learning, or other machine learning algorithms within the simplified version of Overcooked. The goal is to implement a specialized and general-purpose controller which is able to adapt to a range of scenarios and then evaluate its performance against existing results, such as those presented in literature [1].

The reason for studying Overcooked-AI is also to understand how humans and AI can cooperate better, by understanding where cooperation fails, and how to mitigate those failures.

Overcooked-AI is a modified version of the original game that retains the core mechanics and cooperative nature but reduces its computational complexity. It offers a more discrete action space, simplified observation space, and is designed to be computationally lighter. The main differences are that ingredients do not need to be cut, they can be directly added to the pot, and that there are no dirty plates. The pot is also not a different object, and it cannot be taken out of the stove.

Overcooked-AI is used for research purposes because running the full game can be computationally intensive. By using the simplified version, human-AI cooperation research is more efficient.

The main questions that are going to be answered in this paper are as follows:

- How can cooperation fail, and what can we do to prevent these failures?
- Do specialized, hand-authored controllers perform better than those trained with different algorithms, such as population based learning or planning methods?

In other words, this research studies a hand-authored controller designed to collaborate with humans in order to deliver as many soups as possible in the given time. The results are then compared with previous versions of the same controller and with results from other papers that have studied Overcooked-AI. The controller will also be tested cooperating with the researcher himself in order to better evaluate its performance.

2 Background

In this section, we provide some background information in relation to Overcooked-AI, Overcooked, Plate Up! and discuss the techniques employed in designing, creating and evaluating the hand-authored controller.

The investigation of human-AI cooperation has surged in recent years, going beyond the traditional focus on standalone AI agents and extending into dynamic environments like video games, where agents need to cooperate with each other to achieve optimal performance. One particularly promising environment for studying this interaction is Overcooked [2], a cross-platform multiplayer game known for its fully collaborative nature and diverse range of cooking tasks.

The games from the Overcooked series contain many interesting mechanics that focus on cooperation and task distribution that is evenly split between players. These mechanics include cutting ingredients in specific locations prior to cooking, passing food from one kitchen section to another, ensuring orders are prepared in the correct sequence, and managing dishes by washing dirty plates. Some levels also have a dynamic environment, which means that counters or different other objects change their position throughout the level. Each region contains level based mechanics, such as portals, conveyors, cars that can kill the player, people that move and can block some parts of the level.

However, running the complete game poses computational challenges due to the demanding requirements of high-frequency, continuous control from agents. To overcome this obstacle, a simplified implementation of Overcooked that preserves the game's multi-agent mechanics was used, which significantly reduces the computational complexity. This simplified version offers a discrete action space, simplified observation space, and a suitable interface for conducting human subject experiments.

In the context of cooperative games, it is worth mentioning the game Plate Up[3] as well. Plate Up! is another restaurant management game based on cooperation. A representative image is presented in Figure 1. It is very similar to both Overcooked and Overcooked-AI. This is why, in future, some mechanics from this game could be implemented in the simplified version of Overcooked as well.

The main goal is the same: to deliver meals, but the delivery location is dependent on where do the guests arrive. There are of course, many other details added that increase the complexity, such as having to mop the floors in order to gain a speed advantage, being allowed to design the layout of the system in a cooperative manner and having to decide together on the next kitchen upgrades based on current layout. Some recipes involve the generation of food scraps, which need to be taken to the trash in order to clear the space, and then the trash needs to be taken outside of the restaurant. The goal is to be able to stay open for as many days in a row as possible, by improving the kitchen equipment with the money received during the day.

Figure 1: Representative image of the game Plate Up [3]

To achieve the goal of designing and implementing a scripted-AI, a range of Artificial Intelligence techniques, including Behavior Trees, Finite State Machines, Decision Trees, and Utility systems will be considered for the design. While Finite State Machines have traditionally been employed for similar tasks, their scalability becomes a concern as complexity increases[4], leading to the consideration of Behavior Trees.

Behavior Trees are a hierarchical structure of organizing the switching mechanism of an agent [5]. This means that it provides a high flexibility and scalability in the decision-making process.

Artificial Intelligence techniques such as Decision Trees, which can from training data derive a set of rules in order to make decisions [6] will be used. The rules will be derived from observations and mathematical calculations, and will be manually adjusted in order to improve performance. In the final implementation, a small behavior tree has been used which contains four types of behaviours. The scripted-AI switches between them according to the current state of the environment and according to the map analysis made at the beginning. Inside the behaviour, there are a list of conditions which choose to best action.

In addition to performance evaluation, the project will focus on identifying specific human-AI coordination failures within the Overcooked environment. By analyzing these failures, the scripted controller can be developed such that it effectively addresses and mitigates such challenges, improving the overall human-AI coordination.

3 Methodology

The methodology employed in this research builds upon the foundation of the game Overcooked. This section provides an overview of the game's mechanics and details to provide a clearer understanding of the research methods used.

The problem statement of the Overcooked environment is delivering as many soups as possible within a specified time interval.

Overcooked is a two-dimensional game that features two controllers and various types of tiles, including empty tiles, counter tiles, dish dispenser tiles, pot tiles, onion dispenser tiles, and delivery tiles, each with specific attributes.

The players start on an empty tile and are unable to occupy the same empty tile simultaneously. They can move in a specific direction, and their rotation aligns with the direction they moved. If movement is not possible, they will only rotate in the available direction. If two players try to move in the same tile, none of them will and their rotations won't change. Players can interact with objects located in front of them, and they have the ability to hold items such as soups, dishes, and onions. Interaction with a dish dispenser or an onion dispenser generates a dish or an onion in their hand, respectively.

When players with a held item interact with an empty counter, they place the object on the counter for later retrieval. By interacting with a delivery tile while holding a soup, the soup is delivered, resulting in an increase in score based on the recipe used. For this experiment, a soup made of three onions is assigned a score of 20 points. To prepare the soup, players must carry three onions to an empty pot and initiate the cooking process by interacting with the pot. The soup requires a specific number of time steps to complete, depending on the layout of the game.

The goal of this research is to create a controller that can effectively implement the mechanics of the game and collaborate with other players to maximize the number of soups delivered within a given time frame. By addressing this problem, the study aims to explore a strategy for efficient cooperation.

The research has been done using the GitHub repository[7] which contains the implementation of Overcooked-AI. The code was implemented using Python and deployed using Docker.

4 Scripted AI's design

The following section presents the contribution done in this research. It describes how the AI agent was built, including the implementation of a path-finding algorithm for efficient navigation in the Overcooked environment. It also provides details on specific implementation considerations and offers examples of situations where cooperation between human and AI agents may face difficulties.

The AI tries to use a greedy approach in solving the problem. This means that he will try to act mostly depending on the current state. There are four different behaviours the AI can have. These are "Bring Onion", "Pick Up Soup", "Bring Plate" and "Deliver Soup".

The default behavior for the AI agent is the "Bring Onion" behavior, which is applied when there is no need for other behaviors. In this behavior, the AI's actions are primarily determined by the item it is currently holding. If the AI is holding an onion, it calculates a score for each tile to determine the optimal placement of the onion, prioritizing pots that already have onions in them, and then attempts to place it on the tile with the highest score. If the AI is not holding an onion, it calculates a score for each tile to determine the optimal location to take an onion from, and takes it from the tile with the highest score. This score is mostly based on the distance to the location, and if it's possible to accomplish the goal. If the AI is holding a dish, it tries to drop it on the nearest counter. Currently, there is no way for the AI to be in this behavior while holding a soup, as holding a soup triggers the "Deliver Soup" behavior.

Figure 2 provides a concise illustration of the behavior selection process.

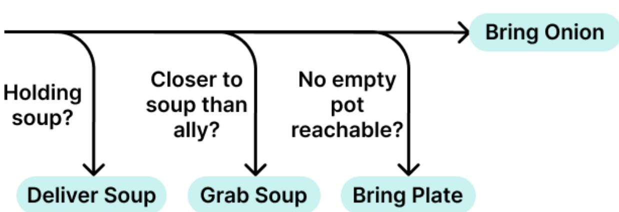

Figure 2: Image representing the scripted AI's behavior selection

The "Grab Soup" behavior which is the same as "Pick Up Soup" behavior is utilized when there is a soup that has finished cooking and is ready to be picked up, provided that the other controller cannot pick it up faster. The decision to pick up the soup is determined by calculating the paths from both controllers to the pot, considering any additional paths required (such as dropping an onion). If there is a tie in distances, the controller's ID is used as a tiebreaker. When the AI is in this behavior, it follows a specific procedure to pick up the soup. First, it drops any onion it is currently holding, then it picks up a dish, and finally, it picks up the soup.

The "Bring Plate" behavior is highly dependent on the specific map layout. In the current maps, it only occurs during Forced Coordination map. This behavior is triggered when there is no plate within reach of a pot, but there is a plate reachable by the controller. In this scenario, the controller attempts to pick up a plate and drop it within reach of the other player and the pot.

The "Deliver Soup" behavior occurs when the controller is holding a soup. In this behavior, the controller will simply deliver the soup to the nearest reachable delivery tile.

The distances in the game are calculated using a slightly modified version of the Lee Algorithm, which is a variant of the Breadth First Search algorithm. This algorithm is well-suited for finding the shortest path [8] and provides satisfactory results for the relatively simple maps in the game. While other solutions, such as Cooperative Pathfinding [9], were considered for this research, their implementation complexity was deemed high compared to the benefits, given the relatively low difficulty of the maps.

A minor adjustment was necessary due to the rotational mechanics in the game, which can affect the overall distance calculation. As illustrated in Figure 3, despite both paths having the same length (2), the red path offers a more advantageous end rotation. This is because the interaction can be performed immediately after reaching the end, eliminating the need to first rotate and then interact, which would cost one additional action. This means that, in this scenario, the red path should be preferred over the black path.

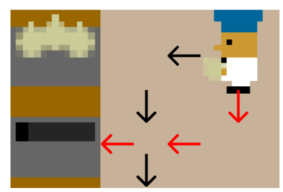

Figure 3: Image showing two different paths with the same length, but different end orientations

To summarize, in this section, I have discussed the implementation of the algorithm and the pathfinding algorithm in the context of Overcooked-AI. The algorithm consists of several behavior modules, which dictate the model's actions based on the current game state. The pathfinding algorithm, based on a modified version of the Lee Algorithm, is used to calculate distances and determine optimal paths. The controller's performance was evaluated through testing with both the AI playing against itself and the AI cooperating with a human player. These tests will be further discussed in the next section.

5 Experimental Setup and Results

The setup can be done using the instructions from the following commit [10]. After installation, a Docker instance can be used in order to host a local version of the Simplified Overcooked game. After that, RandAI needs to be selected in the player spot where the Scripted AI will play. In order to have the scripted AI's behaviour, the file Game.py in the docker instance needs to be replaced with the one that can be found and downloaded from GitHub [11].

The first experiment done in order to asses the Scripted AI score is to compare it when cooperating with itself. This can be done by selecting RandAI both as Player 1 and as Player 2. The map will be automatically loaded into the Scripted AI memory, and he will be able to perform logic actions on all of the maps. After testing with 1 tick per AI action and a time horizon of 400 steps (this means 67 seconds in the Docker simulation), the results can be seen in Figure 4. The plot also contains a reference to results obtained in literature, and also results when cooperating with a different version of the Scripted AI. In order to obtain the different version of the Scripted AI, the logic that activated the behaviour of "Pick Up Soup" was activated only if the id was the specified one. This was to test its adaptability to an edge case.

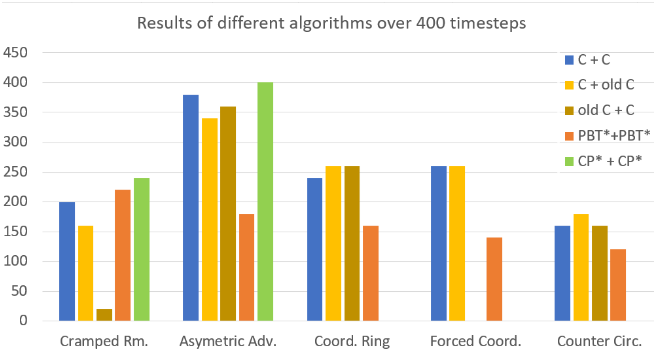

Figure 4: Results of different maps. Reward per soup is 20 points and the total time step is 400 (67 seconds). PBT and CP results are taken from literature[1]. "C" represents the controller, and "old C" represents a version which does never pick up soup, but only delivers onions. CP + CP results are available for only two layouts.

The results can be compared with the ones obtained using the two different methods. The first comparison is made between the scripted AI and an AI obtained using Population Based Training (PBT) from literature [1] which can be found in Figure 5.

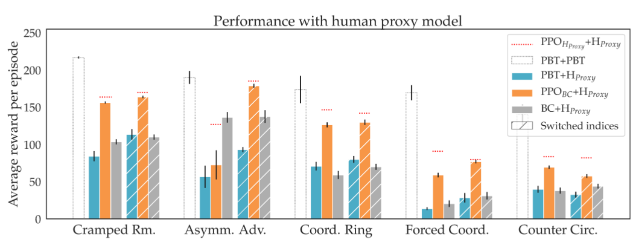

Figure 5: "Comparison between agents trained via PBT"[1]

From the plots, it can be seen that the scripted AI performs overall better than the AI obtained using PBT. The comparison is made using the white lines in Figure 5. The same maximum time was used, which was 400 time steps and the reward per soup delivered is 20.

A map by map analysis can also be made and an image has been inserted for each for easier understanding.

- For the Cramped Room, in Figure 6, PBT performs better than the Scripted AI, sucesfully delivering one more soup than the scripted AI.

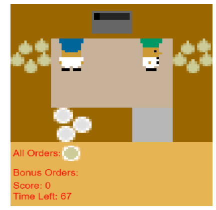

Figure 6: Cramped Room

- For the Asymmetric Advantages, in Figure 7, the Scripted AI greatly outperforms PBT. The difference is of 10 soups. The map might be difficult for PBT because of the open space, and the scripted AI can easily go over this problem using the path finding algorithm. the scripted AI was unable to identify the correct solution in this case, resulting in both controllers delivering soup instead of only the green hat controller, as intended by the map layout. .

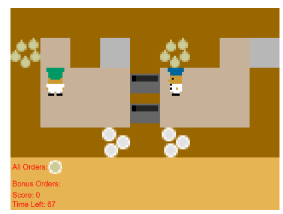

Figure 7: Asymmetric Advantages

- For Coordination Ring, in Figure 8, the difference is of 5 soups, favoring Scripted-AI. Here, the Scripted-AI manages to make use of the counter in middle in a few occasions, most often when another soup is almost finished. The usual movement direction is typically clockwise, although there are instances where it may change direction temporarily. Additionally, there is a rare position on this particular map that can cause the controller to freeze.

Figure 8: Coordination Ring

- For Forced Coordination, in Figure 9, the difference is of 4 soups, favoring Scripted-AI. This map is difficult, because, as the name implies, it cannot be completed at all if one controller does not know what to do. The left controller is responsible for placing onions and dishes on the counter, while the right controller adds onions to the pots and delivering the soups.

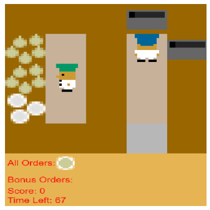

Figure 9: Forced Coordination

- For Counter Circuit, in Figure 10, the difference is of 2 soups favoring Scripted-AI, but this is far from the best outcome. Neither PBT nor the Scripted-AI could find a solution for this map. The solution involves passing onions trough the center of the map. During the tests with the human researcher, the Scripted AI demonstrated cooperation with the human player who knew the correct solution. When playing together, they achieved higher scores than when the Scripted AI played alone.

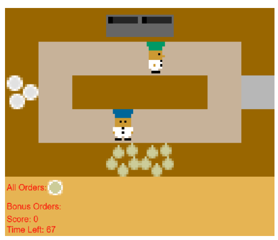

Figure 10: Counter Circuit

It is worth noting that the Scripted-AI utilized the same script across all maps, while the PBT approach employed different controllers for each map and positioning, resulting in two controllers for each map depending on the starting position. The Scripted AI can handle different map layouts, while the PBT algorithms are designed for specific layouts and require separate training for each new map.

Another comparison can be done between the Scripted-AI and the Planning Methods from Figure 11.

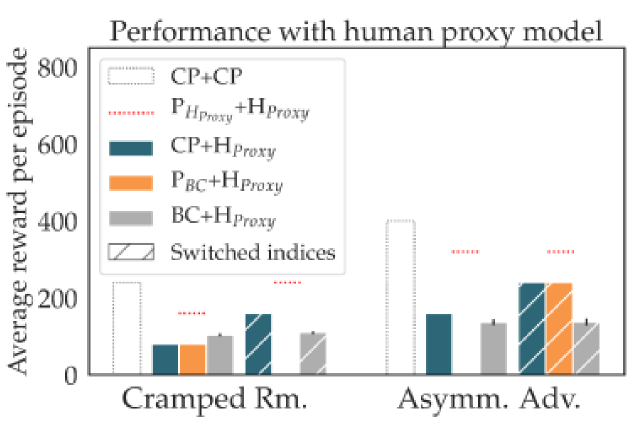

Figure 11: ”Comparison across planning methods.” [1]

Again, because the Scripted-AI plays only against itself, in order to make a comparison we can only look at the white line.

In both maps, the Planning Methods show higher performance compared to the Scripted-AI. It is worth noting, however, that this comparison is limited to these specific maps. The higher scores achieved by the Planning Methods can be attributed to their greater computational complexity and ability to consider future actions.

In contrast, the Scripted-AI exhibits a faster execution but adopts a more immediate and less forward-looking approach due to its greedy nature.

Regarding cooperation with a modified version of itself, the Scripted AI managed to get relatively the same results as when cooperating with itself. This is surprising, because removing completely a behaviour was expected to have a greater effect on the overall score. On the layout ”Cramped Room”, because of the same distance to soup, and the ID being a deciding factor, a freeze situation happened, which resulted in a very low score in one of the configurations.

To conduct a comparative analysis with human players, the scripted AI was tested by interacting with myself. This approach aimed to gather more accurate results and provide insights into the performance of the AI controller.

A specific evaluation method was devised to measure the effectiveness of the scripted AI. Instead of relying solely on the number of soups delivered, which can vary greatly based on the timing of each delivery, a more accurate metric was used. The idea was on calculating the number of time steps required to deliver a fixed number of soups, specifically 10 soups or a total score of 200 points.

This evaluation method ensures a more precise assessment of the AI's performance as it minimizes the impact of minor time step differences in soup production. By considering the total time steps required to deliver a fixed number of soups, the evaluation becomes more consistent and provides a fairer comparison across different scenarios.

For instance, in Figure 12, it can be observed that even though one additional soup may have been delivered, the number of time steps finished. This means that the overall evaluation is heavily influenced by a single time step. The new method allows for a more reliable assessment of the AI's performance without exaggerating the significance of small time step variations.

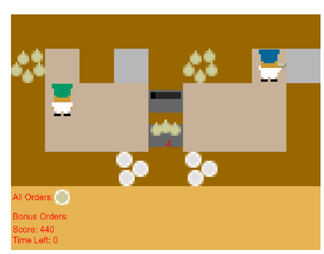

Figure 12: Example where one extra turn could mean one more delivery. Player with blue hat is holding a soup, but timer ran out.

Tests were conducted using the previously mentioned method, involving interactions between the Scripted-AI and itself, as well as between the Scripted-AI and the researcher of this paper. These tests were made possible by the Docker deployed version of the game, enabling human interaction with the AI.

During the testing process, the researcher made efforts to interact with the Scripted-AI to the best of their ability. It is important to note that the results may be influenced by a learning factor, with potentially higher scores observed in the later maps, as the testing followed the sequential order of map appearance. Additionally, due to the absence of a countdown, the human researcher may have accidentally skipped some initial timesteps, as the game starts immediately upon pressing the start button. Considering that each timestep corresponds to 1/6 of a second, the likelihood of missed timesteps is significant. However, the implementation of action queuing played a role in minimizing the overall number of lost timesteps resulting from human errors during the tests.

The results can be seen in Figure 13. Each of the five columns represent one map layout, and the three different colours represent the configurations.

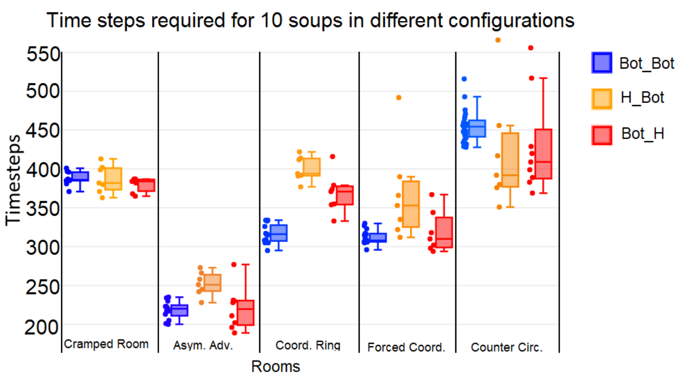

Figure 13: Whisker plot of time steps required to deliver 10 soups on different maps and with different configurations. Bot_Bot means that scripted AI played against itself, while H_Bot means that player one was human-controlled and the second player was controlled by the scripted AI. Similarly, Bot_H means that the scripted AI controlled player one while the other player was human-controlled. In the maps previously shown, player one has a blue hat while player two has a green hat.

A significant imbalance is evident in maps such as Forced Coordination and Asymmetrical Advantages, where players are positioned on two different sides of the map. In these maps, there is a notable difference in performance. However, on the remaining maps, the mean performance is relatively consistent. Coordination Ring may appear to be an exception, but the mean scores are still relatively close.

The map that seems to be the easiest is Asymmetrical Advantages, while Counter Circuit poses the greatest difficulty.

It is worth noting that, uniquely in Counter Circuit, the overall score is better when playing with a human. This observation could be attributed to the human player's ability to efficiently pass onions over the counter.

In Cramped Room, no noticeable difference was observed when the Scripted AI was paired with a human player compared to when it played with itself. There was no significant difference when one of the controllers never delivered the soup, indicating that the controller is capable of effectively cooperating in such situations.

In general, the Scripted-AI shows good performance when paired with the same version of itself. Particularly in time-sensitive scenarios or when adapting to dynamic maps is needed, the Scripted-AI might be the best solution.

6 Responsible Research

In the context of the Overcooked-AI research, it is important to address the ethical aspects associated with the development and deployment of the AI system. The research practices used in this study include considerations such as data privacy, fairness, and transparency.

Regarding data privacy, the collection and use of data from human subjects adheres to ethical guidelines and ensure informed consent. No identification data was collected, and the only test subject of this research was the researcher himself.

Fairness is a critical aspect of AI research. The development of the scripted AI in Overcooked aims to ensure that it does not discriminate against players or exhibit biased behavior. The AI provides equal opportunities and experiences to all players, regardless of their background or characteristics. The AI cannot make any difference between players based on other factors other than in-game actions.

Transparency plays a vital role in the ethical conduct of research. The implementation details of the scripted AI are documented and are available to the research community. They are hosted on GitHub [11]. This allows for easy replication, and validation of the findings.

Moreover, in terms of reproducibility, efforts have been made to provide detailed documentation of the methodology, algorithms, and techniques used in the development of the Overcooked-AI. This enables other researchers to replicate and validate the results.

Additionally, this research explores the cooperative aspects of human-AI interaction. By studying and improving the performance of AI agents in cooperative tasks, the aim is to enhance collaboration and teamwork between humans and AI systems. This has potential implications in various domains, including gaming, human-computer interaction, and multi-agent systems.

By emphasizing ethical considerations, fairness, transparency, and reproducibility, this research tries to ensure that the development and deployment of the scripted AI for Overcooked aligns with responsible and ethical practices. Through these efforts, the study aims to contribute to the advancement of AI technologies while maintaining ethical standards.

7 Discussion

The limitations of this research primarily rise from the programming of the Scripted-AI. It is important to acknowledge that the script may contain bugs or take suboptimal decisions, which could affect its performance and behavior. Additionally, the evaluation of the Scripted-AI was limited to the provided maps in the Overcooked environment. Therefore, its effectiveness and adaptability on other maps or in more complex scenarios remain uncertain.

To address these limitations, further analysis and testing are necessary. Additional experiments could be conducted to evaluate the performance of the Scripted-AI on different maps with varying levels of complexity. This would provide a more comprehensive understanding of its capabilities and identify areas for improvement. It would also be helpful to conduct debugging and refinement to enhance the Scripted-AI's functionality and address any issues or inconsistencies that may arise.

While the current findings show the performance of the Scripted-AI in the given context, it is important to recognize the limitations and the need for further research. By addressing these limitations and refining the programming of the Scripted-AI, future studies can increase our understanding of its capabilities and explore its potential in a larger range of scenarios and environments.

8 Conclusions and Future Work

In conclusion, this research successfully developed and analyzed a Scripted AI controller for the simplified game Overcooked. The AI demonstrated its cooperation capabilities through different testing methods and comparisons to other AI techniques from the literature.

Using a Behaviour Model and Decision Trees, the Scripted AI effectively navigated various map layouts and achieved competitive scores on each tested map. In comparison to agents trained via Population Based Training from the literature, the Scripted AI showed higher performance. However, when compared to Planning Methods also from the literature, its scores were lower.

In collaborative scenarios with human players, the Scripted AI's performance generally decreased, except for a specific map where the human's assistance improved the overall score. This suggests that while the Scripted AI usually performs better on its own, it can strategically cooperate with humans in specific circumstances to achieve higher scores.

Future work can involve refining the Scripted AI controller, exploring alternative AI approaches in this environment, and addressing any identified bugs or issues in the AI's code. Continued improvement, thorough testing on different map layouts, and evaluating performance with multiple players can increase the understanding of the Scripted AI's capabilities.

Furthermore, incorporating additional mechanics such as the ability to throw food, managing food waste, dish-washing, utilizing specific cutting counters for slicing objects, supporting more than two controllers, implementing communication between players, and introducing the concept of overcooking when a soup is left on a pot for an extended period can significantly increase the complexity of the tasks and contribute to more interesting and challenging scenarios where effective cooperation and good decision-making become crucial.

References

[1] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

[2] Ghost Town Games Ltd. Overcooked on steam. https://store.steampowered.com/app/448510/Overcooked/.

[3] Yogscast Games. Plateup on steam. https://store.steampowered.com/app/1599600/PlateUp/.

[4] Chong-U Lim, Robin Baumgarten, and Simon Colton. Evolving behaviour trees for the commercial game defcon. In *Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part I*, pages 100–110. Springer Berlin Heidelberg, 2010.

[5] Michele Colledanchise and Petter Ögren. How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees. *IEEE Transactions on robotics*, 33(2):372–389, 2016.

[6] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.

[7] Humancompatibleai. https://github.com/HumanCompatibleAI/overcooked_ai.

[8] Robbi Rahim, Dahlan Abdullah, Saiful Nurarif, Mukhlis Ramadhan, Badrul Anwar, Muhammad Dahria, Surya Darma Nasution, Tengku Mohd Diansyah, and Mufida Khairani. Breadth first search approach for shortest path solution in cartesian area. In *Journal of Physics: Conference Series*, volume 1019, page 012038. IOP Publishing, 2018.

[9] David Silver. Cooperative pathfinding. In *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment*, volume 1, pages 117–122, 2005.

[10] HumanCompatibleAI. Merge request 121 from humancompatibleai. https://github.com/HumanCompatibleAI/overcooked_ai/commit/d4c4c9fc90daec9e404603578790890 2542fba44.

[11] Anton Mihai Cosmin. Scripted-ai for overcooked! https://github.com/TheDarkVoice2013/Overcooked-AI/.