A Multi-agent System For KAZ

AASMA 2023 Final Report

Group 04
Tomás Pereira 95682
Maria Campos 95629
Sofia Bonifácio 92559

ABSTRACT

Knights Archers Zombies (KAZ) is a game from PettingZoo where a group of Archers and Knights have to prevent an endless spawning swarm of Zombies from reaching the bottom of the screen. The game ends when either a Zombie reaches the bottom or all the players are killed by colliding with Zombies. We will be addressing this game as a multi-agent coordination problem, where we'll have to find the best way to make a Knight and an Archer coordinate and cooperate in order to achieve the best results possible. To reach the desired solution, we will be analyzing different experiment results built upon a greedy agent: first a simple baseline, afterwards the implementation of social conventions and finally the implementation of roles.

KEYWORDS

Multi-agent system; Coordination; Cooperation; Reactive; Environment; PettingZoo

1 INTRODUCTION

1.1 Motivation and Related Works

This game is a prime example of a multi-agent scenario. Archers and Knights can attack Zombies with long ranged arrows or a close ranged mace, respectively. Knights can hit multiple Zombies at the same time, while Archers can only hit one per arrow. Not only that, Archers have a limit of max arrows on the screen. This makes it so each different type of player has their own strengths and weaknesses, making so that coordination is essential for a good performance in the game.

Given this, the Knights and Archers when controlled by AI have a lot of underlying potential to have somewhat interesting performances. In the untouched PettingZoo game version, the AI that controls these characters acts on random actions, not making use of observations for both attacking and survival measures, meaning that they don't have or don't properly take advantage of any sensors. An example of proper coordination would be for the Archer to focus on far away enemies while the Knight took care of enemies within the Archer's range. But to achieve solutions like this, we need to experiment different rule sets for the agents.

1.2 Problem Definition

In terms of game mechanics, as we have described before, the game abides by the following rule sets:

- Zombies spawn from the top border of the screen and make their way down to the bottom border in random, unpredictable paths.
- Archers spawn on the left side of the lower third of the screen and Knights spawn on the right side of the lower third of the screen.
- There may be any number of Archers and/or Knights.
- Each agent can attack to kill zombies, rotate clockwise or counter-clockwise (15 degrees increments) and move forwards or backwards.
- Knights swing a mace in an arc trajectory in front of their current heading direction and Archers fire arrows in a straight line in the direction of the archer's heading.
- Archer arrows have infinite range.
- The game is over when all agents die or when a zombie reaches the bottom. There is also a maximum number of action cycles to prevent infinite runs. We refer to these as frames.
- Other settings of the game may be customizable, such as agents dying if they hit the lateral borders of the screen, the spawn rate of the zombies and the max amount of zombies on screen.

1.3 Problem Relevance

This game provides a lot of room to develop a multi-agent system in which the agents can achieve the following traits:

- Cooperation: the agents can work together as a team to achieve a common goal. For example, for the common goal of preventing the Zombies from reaching the bottom, the Archers and Knights could kill the Zombies closest to the bottom of the screen in order to avoid a game over.
- Coordination: the agents can manage the inter-dependencies between activities such as not attacking the same Zombie so that the potential for coverage from Zombies is maximized. This could also take into account the attack types of the agents. Since the Knight has a melee attack type, when using the social conventions strategy, he has the highest priority. Meanwhile, the Archer has less priority, since its long ranged attack can reach the whole game field.

1.4 Objectives of the Project

As stated before, our main goal in this project will be to flesh out and model the game as a multi-agent problem to solve various coordination tasks. We will have to ponder which coordination strategies are the best and most effective to last longer in harder iterations of the game.

AASMA Project, IST, 2023, Lisbon, Portugal

https://doi.org/

1.5 Related Work

Coordination based video game agents are already a heavily studied topic in the field of AI, meaning that there may be similar cases that we can base our solutions from and study.

For a simple case of multi-agent based games, there was a tournament called Axelrod's Tournament, which consisted of multiple computer programs competing in a Prisoner's Dilemma game. Various strategies were tested which involved different playing methodologies, such as "being nice", which meant cooperating and never defecting the other player, and "being provokable", which was a strategy following the popular saying "an eye for an eye", that is, returning cooperation for cooperation and returning defection for defection. There were also the "don't be envious" behaviour, which meant the player was greedy and focused on maximising their own score, and the "don't be too clever" behaviour, which meant not deceiving other players at all. Overall, this case provided a lot of test results for different interactions with different contexts.

For a more "in the market" case, there's Age Of Empires II. AoE II is a strategy game in which player or AI-controlled civilizations participate in warfare and society advancement. In the case of AI-controlled participants, the agent could use a variety of strategies for their play style, including the use of a greedy strategy to expand their empires, prioritising territory occupied by the enemy players rather than empty lands. The game also makes use of social conventions, which are implemented through the possibility of making alliances and diplomatic relations with other players or AI Civilizations.

These examples illustrate that multi-agent strategies have had an history of being employed in the context of video-games, whether for practical purposes like in Age of Empires or with the purpose of being studied with the goal of maximising agent results. Axelrod's Tournament is a perfect example of the latter, where the results revealed that the cooperation strategy with more efficiency ended up being the "being nice" one, against what one might have expected.

2 APPROACH

2.1 Environment Specifications

The environment has the standard setup of a PettingZoo "Butterfly" environment, in the sense that it is an Atari-esque Pygame visual space. The environment can act both as an Agent Environment Cycle (AEC), meaning that it acts on sequential agent observations and actions (Static Environment), or as a Parallel Environment, based on Partially Observable Stochastic Games (POSGs), where all agents take actions and observations in parallel (Dynamic Environment). In our case, we have decided to utilize a **Static** environment approach since it is enough to demonstrate the differences in performance between the various multi-agent coordination goals and strategies.

KAZ's environment is seen as **accessible**, since any agent can obtain a complete and accurate up-to-date state of the environment during their observation, and **deterministic**, since actions have single, guaranteed effects. To add to that, since the game space is limited in both actions and fixed two dimensional movement, the environment is also **discrete**. Similar to a lot of game based environments, KAZ's environment is **episodic**, since the different runs of the environment are different and independent from one another.

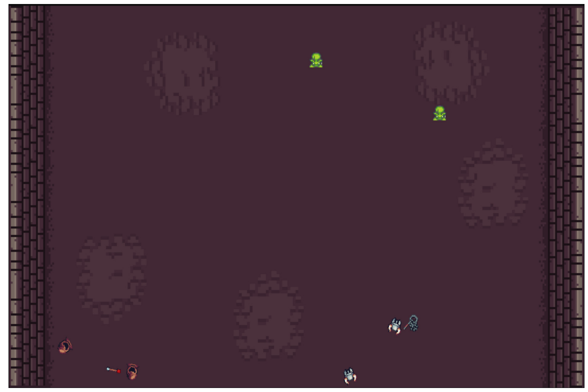

Figure 1: KAZ's environment

2.2 Multi-Agent System

In terms of **sensors**, agents can see the position of zombies and other agents, and in terms of **actuators**, they can choose to rotate, move or attack.

The agents in our system can be seen as **proactive**, as they have goal based behaviour, where they take initiative and recognize opportunities (e.g when a zombie is in range, a Knight can attack it). They also have, as expected, **social ability**, since they are able to communicate and interact with other agents through cooperation and coordination. In terms of properties, agents in this system will be **autonomous**, since they are able to execute actions and infer goals by themselves. This is especially important in the case of the other agent dying, where the agent should be able to still fend off against zombies independently. They will also have **mobility**, as they are able to change their location in the environment, that is to say, in the map.

With this, we can say that in terms of agent architecture, they will be **reactive agents**. This is due to that fact that this type of agent can make the most out of the environment we presented, reacting to unprecedented and unexpected situations from the zombies randomized and constant movement. Other types of agent wouldn't gain any benefits over reactive agents due to the fact that deliberation in this context would put agents at a disadvantage due to the need for fast, spontaneous and decisive actions

3 EMPIRICAL EVALUATION

In this game, there are 3 main metrics that we can infer performance from:

- Amount of zombies killed.
- Amount of agents that died.
- How long the game "run" lasted - measured by the amount of frames that passed.

At first, the number of killed zombies may seem unnecessary since it is expected that the amount is proportional to the length of the game. But, in different contexts and in different settings, these two metrics may have less correlation than what might be inferred at first. So, to this effect, we deemed important to keep them both as separate performance statistics. This ends up reinforcing the fact that it is also of interest to run the game in different settings, to create different contexts of difficulty, such as runs where the zombie density is much higher than normal or where the archer has less maximum arrows shot at once.

We also have to differentiate runs that end due to agents failing to protect themselves from runs that end due to agents letting zombies escape. Tuning their goals towards avoiding these two different "game over" states could be better studied through the amount of agents that died in the run: Did a zombie escape when one agent died? Or while two were alive? Is the most frequent game over state from both agents dying? And so on.

Through analysis of various cases using these 3 metrics, we will be able to infer the strengths and weaknesses of each coordination strategy. The multi-agent system will have the following strategic approaches, all based on **greedy** behaviour:

- Greedy Baseline: No cooperation/coordination between agents
- Greedy Social Conventions
- Greedy Roles

4 IMPLEMENTATION

Throughout the development of the system, we decided on the following set of settings to provide an adequate amount of challenge for the agents being analysed:

- An elevated **spawn rate** with a **maximum amount zombies** on-screen at all times set to 16. This is to induce a high zombie density for a higher difficulty and a more "chaotic" environment for the agents to face.
- **Max Archer Arrows** on screen at all times: 5. This is to reduce the stopping power of Archer agents, as it will be explained further how effective these are.
- **Max Cycles**: 900. This gives the game enough time to provide interesting results without dragging on the episode for too long.

We initially decided on having 2 **Archers** and 2 **Knights** to increase the probability of interesting cooperation and coordination situations and strategies. However, in the middle of implementation, we decided to reduce the number of **Archers** to **1**, due to how their performance in this game completely overshadowed the performance of Knight agents.

4.1 Greedy Policy Baseline

The goal of our baseline was for our agents to have a simple "attack closest enemy" approach. To do so, agents target the Zombie with the smallest absolute distance to them, obtained through the environment's observations.

The most important step in this process was agents having to first correctly rotate towards their target: Their heading direction vector had to be collinear (or close to) to the relative vector to the target they wanted to attack. With that being said, all agents would first rotate the fastest way towards their target if not already aiming at them. Otherwise, they would:

- Attack right way, in case the agent was an **Archer**.
- Move towards the enemy, in case the agent was a **Knight**.

If the target was inside an "in-range" circumference around the Knight, he would promptly attack. This also applied to any Zombie that wasn't the target but was inside the Knight's melee range.

Due to the fact the environment is described through vectors, a lot of reactive checks are based on comparisons with a margin of error. Not only that, all movement actions in the environment (such as rotations and forwards/backwards movements) have set values based on the frame rate of the game and other constants. Sadly, this influenced a lot of precision factors for our agents, but most prominently:

- Made it so that **Archers** would have very poor accuracy against long distance targets, as they would fire at the tip of the vector collinearity error margin and slowly followed the Zombie's movement till it got close enough to hit properly.
- Made it so that **Knights** would occasionally run into Zombies, as the enemy was not inside the attack range margin but the forward movement action put themselves too on top of the enemy.

To mitigate this, we adjusted some constants, but most importantly, made it so the **Archers** aiming system actually shot where the Zombie "was going to go" instead of the Zombie's current position. This greatly enhanced the Archer's accuracy for long and mid ranged targets, but made it very poor against close ranged enemies. In that last case, the Archer would react accordingly and instead aim for the current position of the enemy.

The agents also didn't have any **defensive** reactive behaviours, due to the limited movement actions provided to them. To properly back out of approaching zombies, agents would most likely have to first rotate in order to be able to move away from the threat. However, in most cases, dodging away from close Zombies proved to be ineffective, as it consumed too much time, either making the agents be overrun by even more zombies or accidentally tripping into another one. In the case of the **Knights**, it also generated performance issues, as they achieved very little results without an aggressive behaviour.

4.2 Social Conventions

For our first implementation of cooperation and coordination strategies, we decided to experiment with the use of **Social Conventions**, built upon the **Greedy Baseline**.

In Social Conventions, the Agents would decide their target through the analysis of the 4 closest Zombies to the bottom of the screen, since that is a major and arguably the biggest game over condition for the game. Agents would then target the Zombies assigned to them by the convention, with Knights having priority over Archers. The reason for this higher priority is that Knights can only perform close ranged attacks while Archers can more easily attack any Zombie in the field regardless of distance.

Therefore, the Social Convention logic would be:

- Knight 1 -> Closest Zombie to the Bottom
- Knight 2 -> 2nd Closest Zombie to the Bottom
- Archer 1 -> 3rd Closest Zombie to the Bottom
- Archer 2 -> 4th Closest Zombie to the Bottom

As a bonus, it was also decided that Archer 2 would react according to its initial position before performing any attacks, to move to a declared position on the right side of the screen. This was so

Archer 2 would have a different coverage compared to Archer 1, massively boosting Archer efficiency, as they had a very effective screen-wide coverage of the Zombies.

It was through testing of Social Conventions alongside the initial baseline testing that we noticed how much more effective **Archer** agents were compared to **Knight** agents. Since they only focused on attacking from far away and didn't put themselves into any danger, overall they had excellent performance with very little to no risk of death. We also couldn't increase the game's difficulty as it would prove to substantially reduce the Knight's performance even more. As a solution, we decided to reduce the amount of Archers agents to just **one**. We concluded that KAZ's design was simply biased towards archers, and to have more interesting study results, we finalized the decision of having only 1 Archer in play.

4.3 Roles

For a second cooperation and coordination strategy, we picked Roles as it seemed to be the one that could prove to show the most potential in the context of our game.

For Roles, we once again decided to identify the zombies that were close to the bottom of the screen, as they are the main threats to winning the game. We calculated the distance from each agent to each of the zombies, and, finally, gave each agent the closest zombie to them as their dedicated target. This was calculated with every action taken, meaning that a different Zombie dying in a previous action could still imply an Agent obtaining a new target.

5 EVALUATION AND RESULTS

5.1 First Attempt

After finishing the implementation of all strategies, we decided to compare the results of **50 episodes** of each, utilizing the previously mentioned settings with 1 Archer and 2 Knights. These results are demonstrated by **Figure 2**.

As we can see, the **Baseline** results ended up having overall the worst performance in both kill counts and time survived (frames). Meanwhile both Social Conventions and Roles ended up having very similar performances. We have made the following observations:

- In the **Greedy Baseline**, the Archer never dies. This is due to it having more self-preservation as it always shoots the Zombies closest to it. Meanwhile, in coordination tactics, since it attacks Zombies that are closer to the bottom (as it is the cooperation goal), the Archer sometimes neglects its own survival.
- **Roles** has slightly more **Agent Deaths** than **Social Conventions**, but this difference is easily explained due to the fact that surviving more time may lead to more dead agents in the process.
- The **Knights** achieved a very similar performance, in all cases. This led us to try and optimize their behaviour (as will be explained in the following subsection), since their performance is still very lackluster when compared to the **Archer**

An important detail to note, **Greedy Baseline** most likely has the worst performance due to its most glaring issue: As Agents only target the Zombie closest to them, some stray Zombies which are in farther away positions might never be targeted. This is due to Zombies closer to the Agents' **y axis** taking priority over the Zombies near the edges of the screen, due to how the absolute distance used to choose a target is calculated.

5.2 Second Attempt

Since **Knight** performance revealed to be largely unaffected by coordination strategies, we decided to try and optimise their performance to both rack up more kills and to survive a longer amount of time. To this effect, we made it so Knights would always first rotate to face the target in their attack range, instead of just attacking whenever a Zombie was inside their melee range. We once again ran **50 episodes** of each strategy, and obtained the following results shown by **Figure 3**.

There were four major takeaways from this change:

- **Greedy Baseline** performance **slightly improved**
- **Social Conventions** performance **worsened**
- **Roles** performance **vastly improved**
- **Knight Deaths** decreased over all test cases.

The **Baseline** and **Roles** performance boost can be explained due to the fact that the Knights die less to outlier Zombies in their path. Before, if Zombies came from behind the Knight while they were on their way to the target, it was almost a guaranteed death. Now, the Knight reacts accordingly, first rotating to face the threat and only afterwards attempting to attack said threat, ensuring a higher survival rate. In turn, the Knight surviving more time means that it helps the Archer survive for longer before being overrun by Zombies.

The **Baseline** ended up having less of a performance boost compared to the **Roles** strategy due to the fact that the optimisation didn't handle the Baseline's biggest issue, which was the stray Zombies near the edges of the screen that are never targeted by the closest Zombie targeting policy.

The case of **Social Conventions** is a curious one. In this version of the strategy, we obtained the best possible result in terms of Agent survival across all tests, because the optimisation fixed an issue that was killing off Knights rather frequently: in their long routes from one side of the screen to the other, they often found outlier Zombies which intersected their path and sometimes killed them. However, with the optimisation in mind, it now worsened the effectiveness of the cooperation. This is because the Knight often got sidetracked with its own self defense instead of heading towards the target, which as mentioned before, is one of the two bottom-most Zombies. Ironically, the Knights surviving more often also made it so the Archer could never target these bottom-most Zombies, as the Social Conventions wouldn't allow him to do so.

5.3 Special Attempt - 2 Archers

With the previous results, we decided to check the performance indicators of a game setting where there were **2 Archers** present. The rest of the settings followed the **Non-Optimised Knight** version. The results can be seen in **Figure 4**.

As the graphs reveal, **Social Conventions** is the only strategy that doesn't have a 100% win-rate. This is due to how Social Conventions assigns targets to the agents: there are some cases where some

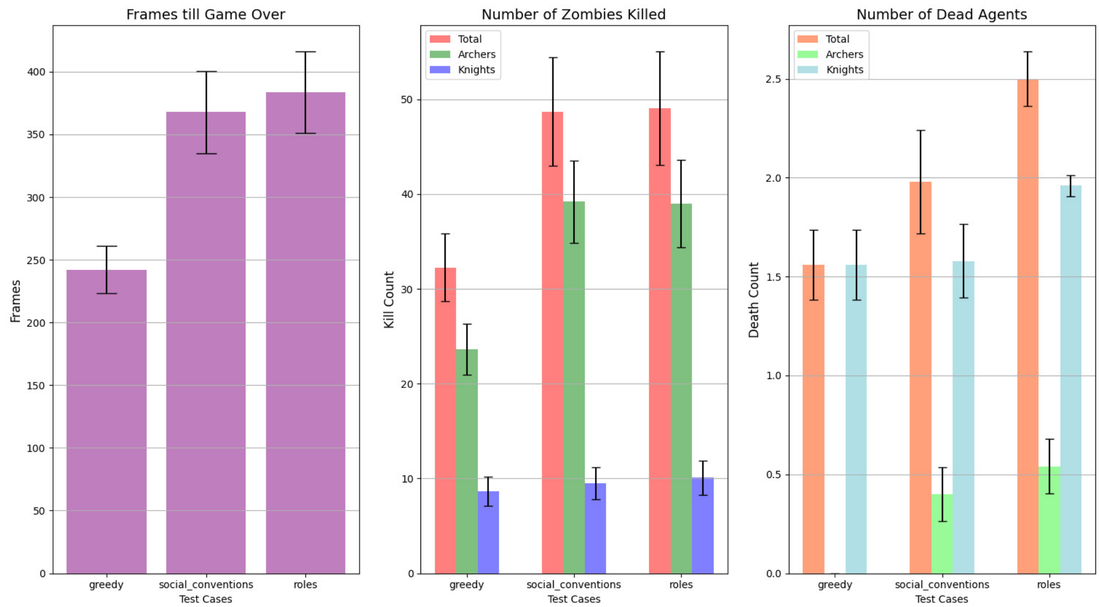

Figure 2: First Evaluation Results

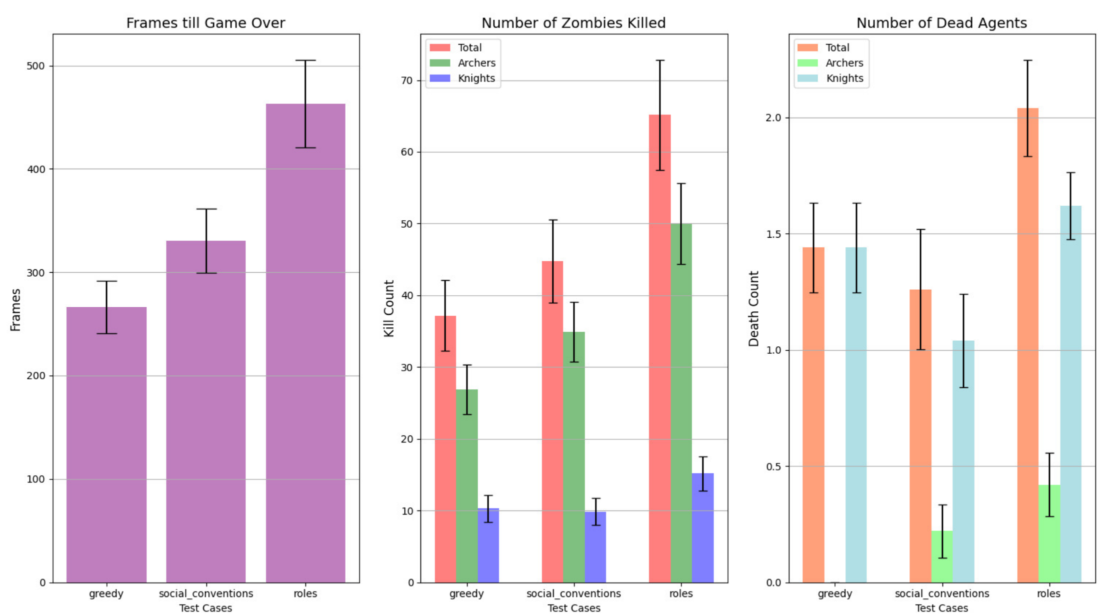

Figure 3: Second Evaluation Results

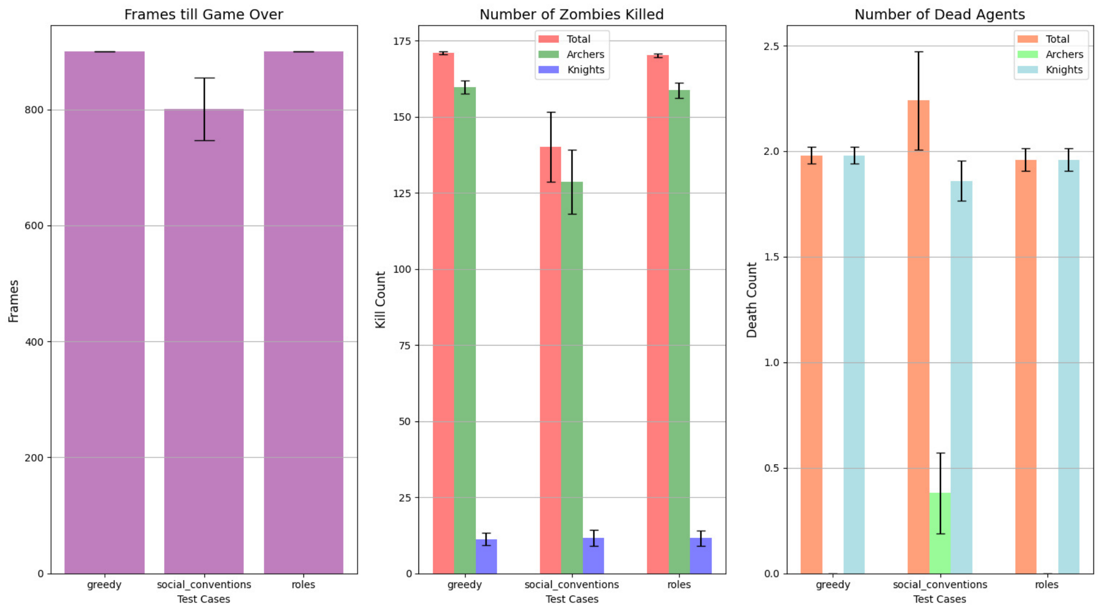

Figure 4: Attempt with 2 Archers Results

Zombies will never be assigned to **Archer** type agents, because Social Conventions' assignments are fixed and an agent won't change their target until said target is killed. As the graphs clearly show, Archers are the ones with the most impact in the performance of the game, so if they don't get a chance to fire towards the most bottom Zombies, it's a big risk of Game Over.

This is also makes it so Social Conventions is the only strategy where Archers are killed, as the priority of an agent when choosing an action is independent of their position and only takes their **type** and **id** into account.

6 CONCLUSION

With all the experimental results we have obtained, one thing was for certain: **KAZ** was a lot more biased towards **Archer** agents than initially expected, making most performance metrics completely dependent on them. Once their performance was limited by game settings (reduced max arrows) and also once we limited their existence to only one agent, we were able to better visualize and infer the benefits of coordination and cooperation.

In **Social Conventions**, the longer the Knights survived, the more opportunity they had to protect the bottom of the screen and therefore the Archer agent. But sadly, the priority system undermined the Archer's potential, inducing multiple situations where Zombies escaped due to Knights being busy and Archer's not being allowed to target said Zombies.

Meanwhile, in the **Roles** strategy, agents were able to make the most out of their current position in order to more efficiently clear out the biggest threats to Game Over: the Zombies near the bottom.

KAZ favors **aggressive** agent behaviour much more than **defensive** behaviour. This is due to how the game mechanics are laid out: Fixed value rotations and 1 axis movement makes it so Agents can't react fast enough to make agent alternatives to attacking matter. Meanwhile, Zombies simply have to move downwards with erratic horizontal movement, meaning that they can easily jump on top of an unsuspecting Agent that is too close. Once again, this greatly favors ranged confrontation for survival purposes, which is the **Archer**'s job, but also makes it so defensive behaviour, in the context of reactive based agents, have a bigger impact on overall performance, as defensive reactions (such as dodging) could easily interrupt and confuse agents when swarmed by zombies.

In the case of KAZ, offense is the best defense. Attacking Zombies prevents them from accumulating too fast, while dodging Zombies would just make them pile up more, which could be especially dangerous when they are closer to the bottom of the screen. The best strategies are the ones that focus on balancing swift kills, such as the closest Zombie to an agent, with effective cooperation goals, such as focusing the Zombies closer to the bottom. The more difficult iterations of KAZ are the ones with an elevated spawn rate of Zombies, making kill throughput that much more important.

Given all of this, we can say with certainty that the most effective strategy was greedy behaviour based **Roles**. Its cooperation goal tackled the most common losing condition, a Zombie reaching the bottom, and it focused on reducing Agent down time by making them efficiently know which Zombie to target, given the full context of all cooperator positions.