Difficulty Scaling In FPS Games Through The Introduction Of Errors In Agent Perception

A. E. Radulescu; A. Guest*
alex.radulescu13@yahoo.com; a.guest@yorksj.ac.uk
York St John University, York

The player experience in first-person shooter games is affected by the skill and predictability of the computer controlled enemies (agents). The performance of the agents is based on the algorithms controlling decision making and the data which is used to make decisions.

This study examined the affects of degrading the quality of data through delay and noise to mimic perception flaws in the agents. A simulation was created in which two agents compete in an arena based first person shooter game. An agent with perfect information was used as a baseline to evaluate the affects of a) delayed data, and b) data distorted through the introduction of noise.

The study used Spearman's rho test to find a high negative correlation (-0,709 for overall agent accuracy and -0,539 for agent precise headshots) between agent performance and the amount of noise distortion of data. Little to no correlation (-0,83 for overall agent accuracy and -0,68 for agent precise headshots) was found when using delayed information.

We conclude that introducing a scalable level of noise to the data used by agents to make decisions provides a simple method of scaling the performance of agents and thus a simple method to scale the difficulty of first person shooter games.

Keywords: Video Games, Player Experience, First Person Shooters, Noise Data, Difficulty Scaling

Introduction

Artificial intelligence in video games plays a crucial role in making the player's experience as immersive as possible by being implemented into realistic agents, friendly or enemy, as well as building mechanics (Machado T. et al, 2018) and storytelling (Kreminski M. et al, 2020).

This study focuses exclusively on enemy agent programming, current techniques and behaviours, as well as limitations and possible further research and developing methods. Due to the complexity of today's games and the quick evolution of technology, the most used AI solutions have become standardized game patterns. This in turn creates a negative effect on the game due to the players already adapting to the AI system, therefore limiting or entirely removing the difficulty factor of enemy agents.

Literature review

Difficulty scaling has become a really useful concept when designing realistic and fair AI for video games. Usually scaling has a multitude of game parameters it can affect, such as rewards, level generation and most importantly the enemy difficulty. It is this adaptability that the AI uses to match the player's skill, either by becoming better or by toning down, that makes the player experience more enjoyable (T. Bruny´e et al, 2005).

When it comes to difficulty adjustment in video games, a strict and levelled approach should be taken, so the AI needs to increase or lower its difficulty in steps. This is done to ensure the player does not constantly overpower the enemy, leading to boredom, nor does the enemy overpower the player, leading to the creation of a stressful and unfair environment (S. Demediuk et al, 2017).

For difficulty scaling to not pose any issues when introducing new AI behaviours it is important to have a central system that holds information about the player. This system, dubbed "Information Manager" (Othman et al., 2012) can hold a multitude of parameters such as player position, direction of movement, etc. In Othman's paper it is discussed how this intermediary system between the AI sensors and decision making system can be used to alter behaviour without changing the behaviour responses directly.

Methodology

Setting up the experiment consists of creating the base AI system, the fully omniscient solution which has a list of parameters that is being sent to the information manger described above. This system will also serve as the benchmark for testing the other two AI versions. The modifications the other versions will have are strictly related to the information manager, and not the behaviour script, therefore the only difference between the systems is the integrity of the data.

The behaviour of the AI consists of finite state machine that has 4 states: "Find Other Cover", "Chase Enemy", "Attack Enemy" and "Run to Cover". The system uses probabilities in some of its calculations, and a bit of randomness is added for more natural behaviour, however the calculations are heavily influenced by the data received from the information manager.

The first modification of the basic system introduces scalable noise into the data values of the information manager. It takes normal values and returns them higher or lower depending on the noise scale. Those modified values include: opposing agent's position, distance to it, the positions of other covers as well as a value modifying the aim accuracy.

The second modification consists of adding a scalable time delay between the game information required by the AI being sent, and the receiver part of the system. The agent would basically receive the with a delay and react to the outdated stimuli, potentially simulating hesitation or slow reflexes.

The information manager is a script that controls both AI data modification methods and has methods for getting the modified data and sending it to the agent. For registering the results of the experiments a battle detail registration system is used, which is constantly being updated with information from a current experiment round.

The experiment itself is using an arena for a 1 versus 1 battle between one of the modified agents, with noise or delay, and the omniscient system. The agents will chase and shoot at each other around covers until one of them is eliminated, moment when the round automatically ends and the results are saved into a spreadsheet. This process repeats a total of 399 times for each alteration method, therefore, in total, there are 798 rounds where the agents fight.

Results and discussion

After running the noise and delay experiments a total of 399 times each, the results have shown early on that due to the broad range of noisy values the AI is processing, the experiment duration is also affected, contrary to delayed information where the duration has been mostly constant. After running normal distribution tests (Kolmogorov-Smirnov and Shapiro-Wilk) on the data it was determined that the data was normally distributed in both cases therefore the experiments produced consistent results.

Using Spearman's rank correlation coefficient, the correlation between the noise affected agent's accuracy and the noise scale is high negatively. While it is not perfectly negative, the -0,709 correlation value offers a clear insight that introducing noise in an agent's data can be scaled and can offer control over the agent's performance by simply modifying the noise amount. Almost no effect has been found on the delayed data, with results being -0,83 and -0,68 in the cases described above, therefore suggesting no real scalable difficulty was achieved using this method.

In addition to the control of agent performance, introducing noise in data has created some mentionable emergent behaviours not previously discovered. One of those would be that as the noise value increased, so did the likelihood of the agent running and shooting at the same time, which made for some intriguing gunfights between the two agents.

Another downside of being affected by noise would be the unresponsiveness of the agent being shot in the back while feeling to cover. This unexpected behaviour can definitely be seen as a weakness in a real game situation, making the AI seem weaker as the noise scale lowers the difficulty.

Finally, the delay data experiments have created emergent behaviours too, one identical to the noise affected agent: shooting while running, which actually helped more in this case than the noisy one. Furthermore, a tendency of the agent to stop at a cover spot and simply camp was noticed, but this was due to the amount of delay between the agent selecting a new cover position, and the agent actually receiving that data. However, the more interesting events occurred when the unaffected enemy would randomly pass by and a fight would ensue, stopping the camping behaviour.

Conclusion

The thorough analysis and resulting information from those experiments have helped meet some of the aims of this research paper, mostly in the area of manipulating information using noise, but using delay in updating information has also proved, if not entirely useful, at least intriguing.

The study's aims regarding noise information alteration have been met, with results indicating that the approach taken combined with rigorous testing did provide a meaningful correlation between an agent's negative performance and increasing the noise in information. The variety of emergent behaviour that resulted from feeding the agent noisy data indicates that developing agents in this way and testing them under different conditions may present even more unexpected results, therefore exploration in this field may be worth pursuing.

As opposed to the successful conclusion of the noise experiments, the analysis of the delay tests has provided little insight on this approach, with results showing no correlation between delayed information and agent performance, however some emergent behaviour noticed has suggested that further research and a modified implementation in that area could be useful and maybe even provide meaningful results.

Finally, agent performance in video games has a big impact on players, therefore the further exploration of new and interesting difficulty scaling methods is a necessary step for evolution in this field. Further experiments might even include new agent actions, attributes as well as new testing environments and last but not least, the ability to test agents in real time games, using real participants for accurate feedback.

References

1. Othman, Nasri, et al. 'Simulation-Based Optimization of StarCraft Tactical AI through Evolutionary Computation'. 2012 IEEE Conference on Computational Intelligence and Games (CIG), 2012, pp. 394–401. IEEE Xplore, https://doi.org/10.1109/CIG.2012.6374182.
2. S. Demediuk, M. Tamassia, W. L. Raffe, F. Zambetta, X. Li and F. Mueller, "Monte Carlo tree search based algorithms for dynamic difficulty adjustment," 2017 IEEE Conference on Computational Intelligence and Games (CIG), 2017, pp. 53-59, doi: 10.1109/CIG.2017.8080415.
3. A. L. Alexander, T. Bruny´e, J. Sidman, and S. A. Weil, "From gaming to training: A review of studies on fidelity, immersion, presence, and buyin and their effects on transfer in pc-based simulations and games," in The interservice/industry training, simulation, and education conference (I/ITSEC), NTSA, Orlando, Florida, 2005.
4. Kreminski, M., Dickinson, M., Mateas, M. and Wardrip-Fruin, N., 2020. Why Are We Like This?: Exploring writing mechanics for an AI-augmented storytelling game.
5. Machado, T., Gopstein, D., Nealen, A., Nov, O. and Togelius, J., 2018, July. Ai-assisted game debugging with cicero. In 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.