ABSTRACT

Games have long been a popular area of AI research, and become a fast growing software industry since 1990s. Video and symbolic games have become a facet of our daily lives and a major way of entertainment. Despite the fast expansion, research on intelligent games with learning function is still in its initial phase. A game that learns to evolve through interactive actions and plays will make the game more challenging, more attractive, and demand less for labor-intensive script design and planning when developing a game. A major hurdle to implement game learning is the game complexity due to high dimensional, numerical and statistical characteristics of the sensing and action space when simulating a virtual world with the high-realism. The recent research shows that computational intelligence techniques, characterized by numerical learning, optimization and self-organization, may provide a powerful tool set to solve the difficulties in game learning.
The objective of this project is to conduct literature survey on AI games with various learning functions, and develop a simple learning scheme for AI games.

In the first phase, the main areas of AI games will be investigated, including major academic conferences that focus on AI games, and published books.

In the second phase, the task is to investigate the main methodologies used for enhancing game intelligence, including traditional AI methods and newly developed computational intelligence methods. Both first person shooting games and real-time strategy games will be considered.

In the third phase, the focus of investigation will lay on the learning ability and optimization technology in AI games. The feasibility of Q-learning, Bayesian learning, statistical learning, as well as system state space based iterative learning, will be explored.

**Acknowledgements:**

I would like to thank Dr. Xu Jian Xin for being my mentor and guiding and motivating me throughout the entire project in every possible way. I would also like to thank him for providing timely help and advice on various issues that arose from time to time.

**Introduction:**

Game Artificial Intelligence (Game AI) refers to the techniques used in computer and video games to produce the illusion of intelligence in the behavior of non-player characters (NPCs). Artificial intelligence in games is typically used for creating player's opponents. If the computer opponent always does the same thing or is too difficult or too easy, the game will suffer. The main goal of Game AI is to not beat the player, but to merely entertain and be challenging. Advanced AIs are not always fun as they end up making the game much harder. It is essential at this point to understand the difference between Game AI and the AI in the academic field. Game AI is not totally on intelligence and shares very few of the objectives of the academic field of AI. Real AI addresses fields of machine learning and decision making based on arbitrary data input as opposed to game AI which only attempts to produce the illusion of intelligence in the behavior of NPCs [12] [19].

**PHASE 1**

**History of Video Games and Their Use of AI**

The history of artificial intelligence in video games can be dated back to the mid sixties. The earliest real artificial intelligence in gaming was the computer opponent in “Pong” or its variations, which there were many. The incorporation of microprocessors at that time would have allowed better AI which did not happen until much later. Game AI agents for sports games like football and basketball were basically goal-oriented towards scoring points and governed by simple rules that controlled when to pass, shoot, or move. There was a much better improvement in the development of Game AI after the advent of fighting games such as “Kung Foo” for Nintendo or “Mortal Kombat”. The moves of the computer opponents were determined by what each player was currently doing and where they were standing. In the most basic games, there was simply a lookup table for what was currently happening and the appropriate best action. Enemy movement was primarily based on stored patterns. In the most complex cases, the...
computer would perform a short minimax search of the possible state space would be performed and best action would be returned. The minimax search had to be of short depth and less time consuming since the game was occurring in real-time. The emergence of new game genres in the 1990s prompted the use of formal AI tools like finite state machines. Games in all genres started exhibiting much better AI after starting to use nondeterministic AI methods. Currently, driving games like "Nascar 2002" have computer controlled drivers with their own personalities and driving styles [1] [2] [12].

MAIN AREAS WHERE AI APPLIES TO GAMES:

- Non-Player Character AI
- Decision Making and Control
- Machine Learning
- Interactive Storytelling
- Cooperative Behaviours
- Player Control
- Content Creation [14]

CONFERENCES:

Listed below are some of the world’s major conferences that discuss and concentrate on the growth and development of game AI.

- GDC – Game Developers Conference
- The Annual International Conference on Computer Games
- IJCAI which is the International Joint Conference on Artificial Intelligence
- AAAI Conference on Artificial Intelligence
- The ACM international conference [15] [16] [19]

PHASE 2

Traditional Methods

The traditional method for implementing game AI was by mainly using tri-state state machines. The complexity of the AI that could be implemented was restricted because of the tri-state state machines. Traditional AI implementation methods also predominantly followed the deterministic implementation method. In the deterministic implementation method the behaviour or the performance of the NPC’s and the games in general could be specified before hand and it was also too predictable. There was a lack of uncertainty which contributed to the entertaining factor of the game. An example of deterministic behaviour is a simple chasing algorithm [20].

Current Methods
The current methods which are used for implementing Game AI vary from neural network, Bayesian technique, genetic algorithms, finite state machines and Pathfinding. All these methods are feasible but not applicable in every situation given. Game AI is a field where research is still going on and developers are still perfecting the art implementing all these methods in any situation given.

Pathfinding and Steering - *Pathfinding* addresses the problem of finding a good path from the starting point to the goal, avoiding obstacles, avoiding enemies, and minimizing costs in the game. *Movement* addresses the problem of taking a path and moving along it. At one extreme, a sophisticated pathfinder coupled with a trivial movement algorithm would find a path when the object begins to move and the object would follow that path, oblivious to everything else. At the other extreme, a movement-only system would not look ahead to find a path (instead, the initial "path" would be a straight line), but instead take one step at a time, considering the local environment at every point. The gaming industry has found out that the best results are achieved by using both pathfinding and movement algorithms [1] [4].

![Long-Term Pathfinding and Short-Term Steering](image1.png)

**FIG. 1 Illustration of Long-Term Pathfinding and Short-Term Steering**

Finite State Machines and Decision Trees - Finite State Machines (FSMs) describe under which events/conditions a current state is to be replaced by another. Figure 2 shows an example of an FSM. The boxes represent states, which involve specific scripts, animation schemes, etc.

![Example of a Finite State Machine](image2.png)

**FIG 2. Example and working of a finite state machine**
AI is often implemented with finite state machines (FSM's) or layers of finite state machines, which are difficult for game designers to edit. Looking at typical AI FSM's, there are design patterns that occur repeatedly. One can use these patterns to make a custom scripting language that is both powerful and approachable. The technique can be further extended into a "stack machine" so that characters have better memory of previous behaviours [1] [7].

Neural Networks

Neural networks can be used to evolve the gaming AI as the player progresses through the game. Neural networks act as a way to map parameters from one space, an input space, to another space, the output space. The mapping may be highly nonlinear and depends on the structure of the neural network and how it was trained. The best thing with neural networks is that they will continually evolve to suit the player, so even if the player changes his tactics, before long, the network would pick up on it. Some advantage of neural networks over traditional AI are, using a neural network may allow game developers to simplify the coding of complex state machines or rules-based systems by relegating key decision-making processes to one or more trained neural networks and neural networks offer the potential for the game's AI to adapt as the game is played. The biggest problem with Neural Networks programming is that no formal definitions of how to construct an architecture for a given problem have be discovered, so producing a network to perfectly suit your needs takes a lot of trial-and-error [10] [13] [20].

Genetic Algorithms - Genetic algorithms offer a way to solve problems that are difficult for traditional game AI techniques. We can use a genetic algorithm to find the best combination of structures to beat the player. So, the player would go through a small level, and at the end, the program would pick the monsters that fared the best against the player, and use those in the next generation. Slowly, after a lot of playing, some reasonable characteristics would be evolved. Genetic algorithms (GAs) are one of a group of random walk techniques. These techniques attempt to solve problems by searching the solution space using some form of guided randomness. Another technique of this type is simulated annealing. Larger populations and more generations will give us better solutions. This means that Genetic Algorithms are better used offline. One possible way of doing this is by doing all of the Genetic Algorithm work in-house and then releasing an AI tuned by a GA. By having a GA engine to work on the user’s computer while the game is not being played this can be achieved to a certain extent [3] [20] [21].

PHASE 3

Need for a Learning Ability

Learning A.I. would allow the game to surprise the player and maintain the suspension of disbelief as the systems remains invisible. Many games companies are currently looking at the possibility of making games that can match the player's ability by altering tactics and strategy, rather than by improving the ability of opponents. There are few games in the market currently which can uncover a player's tactics and adapt to them. Even on the toughest difficulty settings of most games most players tend to develop a routine, which if they find using successful, will continue using that so that they win more often than not. What would make it interesting at this point is if the AI could work out their favourite hiding places, or uncover their winning tactics and adapt to them. This is a very important feature as it would prolong game-life considerably.
Central to the process of learning, is the adaptation of behavior in order to improve performance. Fundamentally, there are two methods of achieving this. Either by directly changing the behavior and by testing modifications to it, and indirectly by making alterations to certain aspects of behavior based on observations.

![Learning Agent Diagram](image)

**Fig 3. Illustration of a learning agent and its working with the elements and analyser.**

Here are a few of the problems which modern day learning AI commonly encounters when being constructed:

- *Mimicking Stupidity*
- *Over fitting*
- *Local Optimality*
- *Set Behaviour* [6] [17] [20]

Listed below are some of the learning methods. It is essential to keep in mind that not all these learning methods are feasible at all stages.

**Q-learning**

Q-learning is a type of reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the main advantages of Q-learning is that it is able to evaluate the expected utility of the existing actions without requiring a model of the environment. Q-Learning at its simplest uses tables to store data. This very quickly loses viability with increasing levels of complexity of the system it is monitoring/controlling. This reduces the feasibility of the learning technique. Hence Q-learning is not a very famous implementation for machine learning. One answer to this problem is to use an adapted Artificial Neural Network as a function approximator [18].

**Bayesian learning**
Bayesian learning is a method by which players in a game attempt to infer each other's future strategies from the observation of past actions. We study the long-run behaviors of the players in evolutionary coordination games with imperfect monitoring. In each time period, signals corresponding to the players underlying actions, instead of the actions themselves, are observed. The rational quasi-Bayesian learning process is proposed to extract information from the realized signals. We find that player’s long-run behaviors depend not only on the correlations between actions and signals, but on the initial probabilities of risk-dominant and non-risk-dominant equilibria being chosen. There are conditions under which risk-dominant equilibrium, non-risk-dominant equilibrium, and the coexistence of both equilibria emerges in the long run. In some situations, the number of limiting distributions grows unboundedly as the population size grows to infinity. Bayesian learning is one of the most implemented learning techniques and has been found to be very feasible by the game AI developers [22].

Other learning techniques

Apart from the frequently used Bayesian and Q-learning techniques we also have Statistical and iterative learning techniques which can also be used. But their implementation into games is still on the cards and has not been fully researched yet. There's a place where AI, statistics and epistemology-methodology converge and this is where statistical learning comes into play. Under this AI label we can make a machine that can find and learn the regularities in a data set and then make the AI and the game eventually to improve. Based on the outcome of each action it may be selected or avoided in the future. If the data set is really, really big, and we care mostly about making practically valuable predictions, we can use data mining to solve the problem [5] [9].

Conclusion and Future of Game AI

Looking further into the future, the general perception is that AI will be focused not only on optimizing an NPC’s behaviour, but also on the player’s fun and experience in general. This reaches far beyond the guidance of single NPCs into learning what is fun for the player and shaping/changing the game experience accordingly. For example we can create whole cities and civilizations in a believable way. We can also have deep NPC characters, automated storytelling with dynamic tension and emotion planning for the player. It seems to be a fantastic perspective for AI in games. But one cannot restrict game AI growth just to these fields and close the window to new directions where it can be applied. Games feature many great technology directions, and AI is only one of them. But the advances in A.I are the ones that would fundamentally change the way that games are designed. Learning AI and interactive story telling AI are two of the most promising areas where AI growth in the future can lead to a new generation of games [8] [11].

References

[6] Ponsen, M.; Spronck, P.; “Improving adaptive game AI with evolutionary learning”, Faculty of Media & Knowledge Engineering, Delft University of Technology, 2005