Artificial Intelligence in My Eyes on the Applications to Game Design

Senzota Kivaria Semakuwa, Florence Upendo Rashid

Abstract*—Artificial Intelligence (AI) is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. A computer game is an electronic game that involves human interaction with a user interface to generate visual feedback on a video device. Using of AI in game designing makes exciting playing strategies in game, which keeps player attracted and focused on it. Also the AI in game avoid the monotony of repetition where by the player are provided with exciting opponents, more intelligent creative that dwell the world of their games. In order to give a player good game experience, an AI is implemented to produce the illusionary effect of intelligence augments. Here in we are surveying the interaction of AI technology such as path finding and perception, neural networks, finite state machines, rule based systems and genetic algorithm, in different kind of games like strategy, action, adventure and, role playing. We provide comparison of the surveyed technologies in terms of their usability, efficiency and robustness. The survey results indicate the more interaction of finite states machines technology in game design although may not always provide the optimal solution, but it generally provide a simple solution that works. Furthermore a game object that uses an FSM can also use other techniques such as neural networks. For these advantages FSM can be used in most commercial games designing, for example Enemy Nations and Quake

***Index Terms*— Artificial intelligence, Game, Technologies.**

I. INTRODUCTION

The application of AI in game design now days is becoming more better by implementing the incredible complexity of advanced AI engines which has been developed by the efforts and research of programming crowds[1]. The main advances in the early parts from the next century are, the inclusion of player's interactivity and AI in game design. Currently story line in gaming are no longer consists of only one primary character, it must consist of many and all playing an important role in the conflict resolution of their virtual world [2]. On considering this, the AI should be required to operate non-player characters. Using of AI techniques here, produce the illusion of intelligence in the behavior of non-player characters in computer and video games. Creatures need not to be written so as to react based on a single player, and the realness of their behavior and uniqueness of their tactics will be a more important feature than it has been in the past[3]. Generally the use of AI in gaming is creating a player's opponents, where by an intelligence is applying to rescuing the player from the dullness of reiteration and letting him/her to focus on the interesting aspects of the game.

Manuscript Received on August 2014.

Senzota Kivaria, He is working as a Tutorial Asst. in the Department of Computing Science and Studies at Mzumbe University, Tanzania.

Florence Upendo Rashid She is currently an Asst. Lecturer in the Department of Telecommunications and Computer Networks, School of Informatics, College of Informatics and Virtual Education, University of Dodoma, Tanzania.

Game player is giving a high-level strategic orders, the computer-controlled units take care of detail, while the full details and dynamics of the game is maintained by the computer which control details, rather than lost through abstraction. There are many different AI techniques in use in modern computer games. The most prevalent techniques include Finite State Machine, Path Finding and Perception, Neural Network, Rule Based System and E.Genetic Algorithm. Our survey studying these techniques through describing their usability, efficiency and robustness. These techniques includes features like real-time interrogation of suspects, dynamic movement and richness of behavior in design a good game, which in turn increases the interaction between creatures and their environments [4]. Amalgamation together the generative and reactive planning algorithm is generating the path for creating customized and novel behavior of the characters. This will be changing each time the game is played so that, the character behavior is not limited in ability, so as to provide interest to player [5]. The survey is exploring each of these techniques and their robustness, usability and efficiency to the game designing. Each of the technique will be explained its interaction in game design when reconnoitering the following selected games, Action, Adventure, Role Playing, Strategies and Team sport games. The rest of this paper is organized as follows: In section II we explain milestones in the development of AI in game. The five chosen AI technologies in game design is discussed in section III. Section IV provide the interaction of AI technologies in games where by we are discussing how these technologies turns to these games. In additional the survey result is given in section V, where as a summary table of the robustness, usability and efficiency is included.

II. MILESTONES IN THE DEVELOPMENT OF AI IN GAME

While surveying the milestones in the development of intelligence behavior in computer games, the games which have turned out the evolution of AI should be mentioned. One of the most popular games is of 1990's such as Warcraft- a game developed by the Blizzard studio [6]. It was the first game to employ path-finding algorithms at such as grand scale, for hundreds of units in the games engaged in massive battles. Also SimCity, created by the company Maxis, was the first game to prove the feasibility of using A-Life technologies in the field of computer games [7]. Another milestone turned out to be the game Black and White, created in 2001 by Lion head Studios, in which technologies of computer-controlled characters learning for the first time was used [8]. The role of AI can be mille stoned in the market on different genus of computer and video games. Now days the class of computers with Pentium IV processors, frequencies in the range of 3 to 4 GHz can be considered to let computer games to make use of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

the most advanced and sophisticated technologies of AI such as Finite State Machine, Path Finding and Perception, Neural Network, Rule Based System and Human Behavioral Modeling. Currently the age of Internet and Networking games, AI systems in games have been given new task; a computer player should, in its behavior and strategies of playing be indistinguishable from a real player on the other side of Internet connection [9].

III. AI TECHNOLOGIES IN GAME DESIGNING

The core concept behind AI is decision making [10]. To execute these choices, the intelligent system needs to be able to affect the entities using the AI system. It can be organized to execute in either an "AI push" or an "entity pull" strategy. AI push systems tend to isolate the AI system as a separate element of the game architecture. Such a strategy often takes on the form a separate thread or threads in which the AI spends its time calculating the best choices given the game options. When the AI makes a decision, that decision is then broadcast to the entities involved. Entity pull systems work best for games with simple entities. Apart from the decision making, the AI t needs some way of perceiving its environment. In simpler systems, this perception can be a simple check on the position of the player entity [10]. As systems become more demanding, entities need to identify key features of the game world, such as viable paths to walk through, cover-providing terrain, and areas of conflict

A. Finite State Machine (FSM)

A finite state machine (FMS) is a way of conceptualizing and implementing an entity that has distinct states throughout its life [11]. A "state "can represent physical conditions that the entity is in, or it can represent emotional states that the entity can exhibit. In this example, emotional states are nothing like a true AI's emotional states but predetermined behavior models that fit into the context of the game. Figure 1 shows a common examples of states for an AI system for a game with stealth elements:

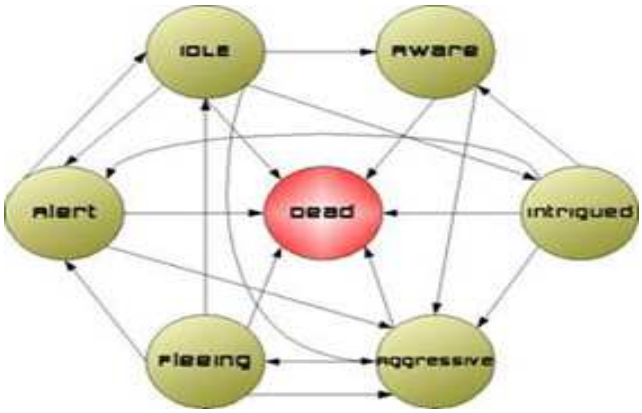

Fig. 1 Layout of the States in a Typical FSM, Where Arrows Represent the Possible Changes in State [11]

Idle, in this state, the entity is passively standing around or walking along a set path. Perceptions are low. Player sounds are not often checked for. Only if this entity is attacked or "sees" a player directly in front of it will its state change to a higher level of awareness.

Aware, this entity is actively searching for intruders. It checks often for the sounds of the player and sees farther and wider than an idle entity. This entity will move to the Intrigued state if it notices something out of place such as open doors, unconscious bodies, or spent bullet casings.

Intrigued, this entity is aware that something is up. To demonstrate this behavior, the entity will abandon its normal post or path and move to areas of interest, such as the aforementioned open doors or bodies. If a player is seen, the entity goes to the Alert state.

Alert, in this state, the entity has become aware of the player and will go through the actions of hunting down the player: moving into range of attack, alerting fellow guards, sounding alarms, and finding cover. When the entity is within range of the enemy, it switches to the Aggressive state.

Aggressive, this is the state where the enemy has engaged in combat with the player. The entity attacks the player when it can and seeks cover between rounds of attack based on attack cool-downs or reloading. The entity only leaves this state if the enemy is killed, if the enemy moves out of firing range go back to the Alert stage, or if the entity dies go to the Dead state. If the entity becomes low on health, it may switch to the Fleeing state, depending on the courage of the specific entity.

Fleeing, in this state, the entity tries to run from combat. Depending on the game, there may be a secondary goal of finding health or leaving the play area. When the entity finds health, it may return to the Alert state and resume combat. An entity that "leaves" is merely deleted.

Dead, in some games, the state of death may not be completely idle. Death or dying can have the entity "cry out," alerting nearby entities, or go into a knocked-out state, where it can later be revived by a medic and returned to a state of Alert. Finite State Machines (FSMs) are used more frequently in computer games than any other AI technique. This is because they are simple to program, easy to understand and debug, and general enough to be used for any problem. On contrary an FSM may not always provide the optimal solution, but it generally provide a simple solution that works. Furthermore a game object that uses an FSM can also use other techniques such as neural networks.

B. Rule Based Systems (RBS)

The most basic form an intelligent system can take is that of a rules-based system. This system stretches the term "artificial intelligence". A set of preset behaviors is used to determine the behavior of game entities. With a variety of actions, the overall result can be a behavior system that is not obvious although there is very little actual intelligence involved [11]. A good example of a rules-based system is a Black Jack dealer. The classic application of this system is Pac-Man.

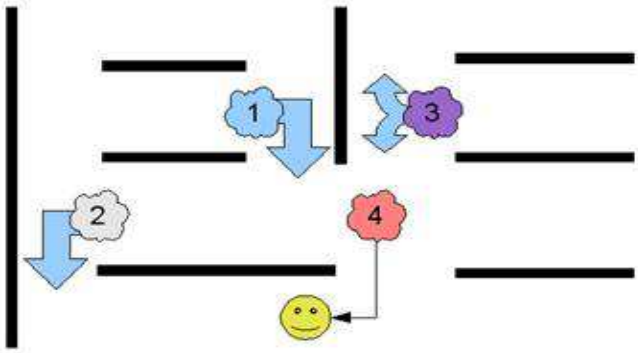

Fig. 2 Visual Representation of the Rule Set Governing *Pac-Man* Ghosts, Where Arrows Represent the "Decisions" that will be Made [11]

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

As in Fig 2 suggests, rules do not need to be hard-coded: They can be based on perceived states or on editable parameters of the entity. Variables such as aggression, courage, range of sight, and rate of thinking can all lead to more diverse entity behavior, even within a rules-based system. Rules-based systems are the simplest structure for an AI. More complex intelligent systems are built upon and governed by a series of conditional rules. In tactical games, rules govern which tactics to use. In strategy games, rules govern build orders and how to react to conflicts. Rules-based systems are the foundation of AI.

C. Path Finding (PF)

Intelligent agents need to identify points of interest in the game world, and then figure out how to get there. All intelligent agents need to have a basic ability to perceive their environment and some means of navigating and moving within the world around them.

Path finding is essentially a solved problem in game development [12]. The core algorithm for path finding has been 'A*' which can be used to discover an optimal path from any two points in a graph. First, set up two lists: a list of nodes that have not been checked and a list of nodes that have been checked. Each list is composed of a position node, the estimated distance to the goal, and a reference to its parent. The lists are initially empty. Next, add the starting position to the unchecked list and nothing as the parent. Then, enter the algorithm:

Select the best-looking node from the list.
If this node is the goal, done.
If this node is not the goal, add it to the checked list.
For each node adjacent to this node:
If the node is non-walkable, ignore it.
If the node is already in a list (Checked or Unchecked), ignore it All
Else, add it to the unchecked list, setting this node as the parent and estimating the distance to the goal.

When entity reaches the goal tile, the path can be constructed by tracing the parents back to the node that does not have a parent. Doing so provides an optimal path that the entity can then follow. Because this process needs to occur only when an agent is either given an order or decides on its own to move, it can really benefit from multithreading. An agent can send a request to the path thread get back a path when it has been discovered without affecting the performance of the AI. For the most part, the system can get results quickly; but in the case of large loads of path requests, the agent can either wait around or just start heading in the right direction. In extremely large maps, the system can be broken down into regions, and all possible paths between the regions can be pre-computed. In that case, the path finder can just look up the best path and return the results immediately. The path map thread can just keep a watch on changes in the map-such as when a player builds a wall-then re-run the path checks as needed. With it being in its own thread, the algorithm can adapt without affecting the performance of the rest of the game. Within the path finding subsystem, multithreading can also improve the performance of the system. This is well utilized in any real-time strategy (RTS) game or a system with a large number of entities each seeking a potentially unique path. Multiple paths can be found simultaneously within different threads. Of course, the system will need to keep track of which paths are being discovered. The same path does not need to be discovered more than once.

D. Neural Networks (NN)

An Artificial Neural Network (NN) is an electronic simulation based on a simplified human brain. In an NN, knowledge is acquired from the environment through a learning process and the network's connection strengths are used to store the acquired knowledge [13].Choosing the variables from the game environment that will be used as inputs is the most labor intensive part of developing an NN [14]. This difficulty is due to the fact that there is wealth of information that can be extracted from the game world and choosing a good combination of relevant variables can be difficult. Also, the number inputs needs to be kept to a minimum to prevent the search space from becoming too large [15]. Therefore, it is a good idea to start with the essential variables and add more if required. Choosing inputs that are poor representations of the game environment is the primary reason for failed applications. NNs are techniques that can be used in a wide variety of applications. Some common uses include memory, pattern recognition, learning and prediction. There are many commercial applications of NNs across various industries, including business, food, financial, medical and health care, science and engineering [16]. Some examples of applications that NNs are being used for are predicting sales, handwritten character recognition for PDAs and faxes, odour analysis via electronic nose, stock market forecasting, credit card fraud detection, Pap smear diagnosis, protein analysis for drug development and weather forecasting. This list illustrates the wide variety of applications that can make successful use of NNs, and how their usefulness is only limited to what can be imagined. The computer game industry is no different from the industries mentioned above in terms of the variety of applications of NNs. A few applications are described by LaMotte, including environmental scanning and classification, memory and behavioral control [17]. The first application, environmental scanning and classification, involves teaching the NN how to interpret various visual and auditory information from the environment, and to possibly choose a response. The second application, memory, involves allowing the AI to learn a set of responses through experience and then respond with the best approximation in a new situation. Finally, behavioral control relates to the output of the NN controlling the actions of the AI, with the inputs being various game engine variables. Also, the NN can be taught to imitate the human player of the game [18]. Basically, an NN can be used to make decisions or interpret data based on previous input and output it has been given. The input can be seen as various games states, similar to that used by a state machine, and the output could be the action to be performed. The important difference is that the current state doesn't need to have been hard-coded. Instead, the NN makes the best approximation that it can, based on the states that it already knows about. This means that the NN will choose an action that would have been performed in a similar state. So far, game developers have been reluctant to allow a game to ship with the learning in NNs and other techniques "switched on", in case the AI were to learn something stupid [19]. Therefore, the developers that have used NNs in their

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

games have not used them for learning, but rather trained them during development and locked the settings before shipping. Finally, the Creatures series of games makes heavy use of Artificial Life techniques, including heterogeneous NNs, in which the neurons are divided into lobes that have individual sets of parameters. In combination with genetic algorithms, the creatures use the NN to learn behavior and preferences over time. In short, NNs are techniques that can be used for a wide range of applications in many different environments. Several commercial games have used this technique successfully, with the most recent and prominent being the game Black & White. This technique's flexibility means that it has the potential to be applied in a wide range of situations in future games. Therefore, it is likely that NNs will play a bigger role in commercial games in the near future.

E. Genetic Algorithm (GA)

A Genetic Algorithm (GA) is an AI technique for optimization and machine learning that uses ideas from evolution and natural selection to evolve a solution to a problem [20]. A GA works by starting with a small number of initial strategies, using these to create an entire population of candidate solutions and evaluating each candidate's ability to solve the problem. Gradually, more effective candidates are evolved over several generations until a specified level of performance is reached [21]. The possible applications of GAs are immense. Any problem that has a large enough search domain could be suitable [22]. Traditional methods of search and optimization are too slow in finding a solution in a very complex search space. However, a GA is a robust search method requiring little information to search effectively in a large, complex or poorly-understood search space. GAs are also useful in nonlinear problems [23]. There are many applications that can benefit from the use of a GA, once an appropriate representation and fitness function has been devised. An effective GA representation and meaningful fitness evaluation are the keys to the success of GA applications. The appeal of GAs comes from their simplicity and elegance as robust search algorithms, as well as from their power to discover good solutions rapidly for difficult high-dimensional problems. GAs are useful and efficient when domain knowledge is scarce or expert knowledge if difficult to encode to narrow the search space, when no mathematical analysis is available and when traditional search methods fail [24]. GAs have been used for problem solving and modelling, and applied to many scientific, engineering, business and entertainment problems [24]. Also, GAs have been extensively explored by academics. However, they are yet to become accepted in game development. They offer opportunities for developing interesting game strategies in areas where traditional game AI is weak. For example, a GA could be used in a real-time strategy game to adapt the computer's strategy to exploit the human player's weaknesses. This GA would need to consider things like how the player's base is set up, how well they can cope with multiple engagements, unit mobility and combined force flexibility. A GA could also be used in a real-time strategy to define the behavior of individual units rather than groups of units or the overall strategy [25]. Additionally, a GA could be used in a role-playing game or first-person shooter to evolve behaviors of characters and events [26]. For example, a GA could take the creatures in the game that have survived the longest and evolve them to produce future generations. This would only need to be done when a new creature is needed [27]. Furthermore, GAs could be used in games for path finding, in which the chromosome could represent a series of vectors and the fitness function could be the distance the sum of vectors is away from a target point [28]. In summary, GAs are based on evolution and natural selection and are used for learning and optimization. They are resource intensive and require much time in development and tuning, which does not make them ideal for in-game learning. Generally, the most difficult part in GA development is determining a suitable representation for the solutions. Also, parameters such as population size, mutation and recombination operators and the number of solutions to erase, make parents or keep unchanged can take a long time to tune. Basically, a GA is not a good algorithm to incorporate into a game where time and resources are limited. Unfortunately, this describes most commercial games. However, GAs also have many advantages, in that they are a robust search method for large, complex or poorly-understood search spaces and nonlinear problems. In short, if GAs are to be used in games, they will most likely be evolved before shipping or between games, and it will be a long time before they become widespread in games.

IV. THE INTERACTIONS OF AI TECHNOLOGIES IN GAMES

A. Action Games

Action games involve the human player controlling a character in a virtual environment, usually running around and using deadly force to save the world from the forces of evil or conquering an alien monsters or mythical creatures. In pure action games, AI is used to control the enemies. Action games like First person shooter type games usually implement the layered structure of artificial intelligence system, layers located at the bottom handle basic task like determining the optimal path to the target and the higher levels take care of tactical reasoning and selecting the behavior which an AI agent should assume in accordance with its present strategy. Providing realism in graphics has been the key point of competition for these games; where AI has played a major role as a point of comparison. Recent games have extended the genre so that the human player may be part of a team, including either human or AI partners. In all cases, it is the moment-to-moment reaction of the AI to the human that is most important so that the AI must be tactically savvy with little emphasis on strategy [29]. Latest trend is to use schedule based finite state machines (FSMs) to determine the behavior of the players' adversaries.

B. Adventure Game

Adventure games, and similar kind of interactive fiction, move further from action games, since they do not give importance to armed combat but accentuate more on story, plot and puzzle solving. In these games, players must solve puzzles and interact with other characters, as they progress through an unfolding adventure that is determined in part by their actions. AI can be used to create realistic supporting goal-driven characters that the player must interact with appropriately to further their progress in the game [29]. The majority of these games has fixed scripts and uses many tricks

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

to force the human player through essentially linear stories. However, a few games, such as Blade Runner, have incorporated some autonomy and dynamic scripting into their characters and story line [30]. Two interesting applications of AI to the adventure game category are the creation of more realistic and engaging non Player Characters and maintaining consistency in dynamic storylines.

C. Role playing Game

In role-playing games, a player can play different types of human characters, such as a combatant, a conjurer or a thief. The player does various kinds of activities like collecting and selling items, fighting with monsters, so that they can expands the capabilities and power of their character like strength, magic or quickness, all in an extended virtual world. The Role playing game format also offers similar kind of challenges to the AI developer as the adventure game with some extra impediment due to the amount of freedom assigned to the player. To maintain a story line consistent for these kinds of games becomes a biggest challenge and higher level of sophistication is required in these types of role playing games. Here AI is implemented to take control over enemies similar to action games, partners who travel and adventure with the players and also supporting characters like traveling companion, villagers etc. The massively multiplayer games provide an additional opportunity to use AI to expand and enhance the player to player social interactions. These days major AI research areas on these types of games is to provide human interaction, social intelligence and natural language interfaces to these support characters [31]. Support characters must provide human-like responses, including realistic movement personality, emotions, natural language understanding and natural language generation. In order to do all this, a large range of integrated AI techniques capabilities are required.

D. Strategies Game

In strategy games, the human controls various kind of entities for example military elements like tanks, guns, war machines in order to conduct a battle from a god's eye view against one or more opponents. Strategy games include reenactments of different types of battles: historical, alternative realities and fictional future and mythical). The human is faced with problems of resource allocation, scheduling production, and organizing defenses and attacks [32]. Strategy games on the market today are an even mix between mythical, fantasy and science fiction campaigns; and recreations of historical battles. There are two distinct classes of game in this category which are turn based strategy (TBS) games involve each player taking their turn to move units, order production, mount attacks and so on and real time strategy (RTS) games which take place in real-time with players moving units, ordering production etc. in parallel . AI is used in two roles: to control the detailed behavior of individual units that the human commands, and as a strategic opponent that must play the same type of game against the human. AI in strategy games needs to be applied both at the level of strategic opponents and at the level of individual units. AI at the strategic level involves the creation of computer opponents capable of mounting ordered, cohesive, well planned and innovative campaigns against the human player. This is very challenging as players quickly identify any rigid strategies and learn to exploit them. At the unit level AI is required in order to allow a player's units to carry out the player's orders as accurately as possible. Challenges at unit level include accurate path finding and allowing units a degree of autonomy in order to be able to behave sensibly without the player's direct control. Neural network used to choose the best strategy in a RTS-type game. Based on situation analysis, the network decides how greatly to concentrate on development, arms production, repairs after battles etc. All the parameters required by the game will be provided by the neural network on its output.

E. Team Sports Game

Team sports games have the human play a combination of coach and player in popular sports, such as football, basketball, soccer, baseball, and hockey [33]. AI is used in two roles that are similar to the roles in strategy games, the first being unit level control of all the individual players. Usually the human controls one key player, like the quarterback, while the computer controls all the other members of the team. A second role is as the case strategic opponent, which in this is the opposing coach. One unique aspect of team sport games is that they also have a role for a commentator, who gives the play by play, and color commentary of the game. In a team sports game the strategic opponent might select a play or strategy for the entire team. That strategy defines a role and/or approximate path for each player involved in the play. Game programmers discovered that the A* search algorithm is a powerful and efficient way to calculate these paths in the sport games.

V. SURVEYING RESULTS

The section is discussing the results of the survey on the interaction of AI technologies with the selected games as explained on section III and IV respectively. In our survey we have seen that most of the games there are designed based on FSM and NN technology of AI. The explore shows that in Action games the finite state machines were used to determine the behavior of the players' adversaries. In Strategy games a NN were used to choose the best strategy in a RTS-type game. Based on situation analysis, the network decides how greatly to concentrate on development, arms production, repairs after battles. The FSM is applied to different kind of game because of its simplicity, can be used in conjunction with other techniques, computationally inexpensive and lots of power relative to complexity. On other side the NN followed an FSM by being flexible, non-deterministic and non-linearity. Also these AI technologies can be applied in managing game world, object or characters for FSM, while the NN is applied on memory, pattern recognition, learning, prediction, classification and behavioral control. Table 1. Shows the comparison of the surveyed technology against usability, robustness and efficiency.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Name of Technology	Usability	Robustness	Efficiency
Finite State Machine	-Natural correspondence between states and behaviors -Easy to diagram -Easy to program -Easy to debug -Completely general to any problem	- Can never approach polish and robustness of commercial compilers/debuggers	-Many extensions like: OnEnter, OnExit Timers Global state, substates Stack-Based
Neural Networks	-Complex non-linear functions that relate one or more inputs to an output	-Training can take place before game ships Once fixed, extremely cheap to compute	-One NN control one game of many game -Several NN control one game.
Path Finding	-not user friendly	-Computationally expensive	-is not very efficiency
Genetic Algorithms	-very limited	- Game may have many settings for the AI, but interaction between settings makes it hard to find an optimal combination	-Good at finding a solution in complex or poorly understood search spaces
Rule Based System	-Foundation of AI	- do not need to be hard-coded	-simplest structure for AI.

VI. CONCLUSION

We have surveyed the AI in game designing from its milestone up to date by reviewing and comparing different AI technologies, their robustness, usability and efficiency on their interaction in different games. Our surveying indicates that a game can be or not relying on single AI technologies in anticipation of its reality, although the FSM technologies have been realized to be used more on designing several kinds of games because an FSM is simple to program, easy to understand and debug and general enough to be used for any problem, however may not always provide the optimal solution. The survey show that games are defined by an explicit set of rules and the goal of playing a game is usually defined unambiguously by the games score or outcome. Also games are immensely flexible and vary greatly in complexity from single player puzzles to two-player board games to massively multi-player real-time video games. Furthermore a techniques which developed specifically for game playing may often be transferred easily to other domains, greatly enhancing the scope with which such techniques may be

used. On summing up academically games playing provide an ideal test bed for the development and testing of new techniques and technologies.

VII. ACKNOWLEDGMENT

We thank the Almighty God for helping us writing this survey paper.

REFERENCES

[1] A. Nareyek, “Game AI Is Dead. Long Live Game AI!” IEEE Computer Society, January/February 2007, Vol. 22, No. 1.
[2] A. Nareyek, “Artificial Intelligence in Computer Games—State of the Art and Future Directions,” ACM Queue, vol. 10, 2004, pp.58–65.
[3] B. Tracy. “Game Intelligence AI Plays Along”. Computer Power User. Volume 2, Issue 1. January 2002. pp 56-60.
[4] A. Nareyek, “Computer Games—Boon or Bane for AI Research?” Künstliche Intelligenz, vol. 18, no. 1, 2004, pp. 43–44.
[5] S. Bandi, M. Cavazza, I.Palmer. “Situated AI in Video Games”. University of Branford, 2009.
[6] M.Morhaime, A. Adham, F. Pearce, “Sylicon and Synapse”, Feb 8, 1991 Blizzed Entertainement, Inc.,
[7] W. Wright, “SimCity”, 1989, Maxis.
[8] K.Greg, "Black & White Review". 30 March 2001, GameSpot.
[9] A.Jakulin, “Artificial Intelligence in Games:” 2003, http://www.stat.columbia.edu/~jakulin/FT,2003.
[10] Z. Cai, G. Xu , Artificial Intelligence: Principles & Applications 4th Edition, , Beijing 2010 Tsinghua Univ. Press, Reprinted 2013
[11] M. Tim Jones. Artificial Intelligence System Approach, John and Bartlett, 2009.
[12] D. Keheo. “Designing Artificial Intelligence for Games part 1”, Feb 9, 2012, Available: https://software.intel.com.
[13] D. Keheo. “Designing Artificial Intelligence for Games part 2”, Feb 9, 2012, Available: https://software.intel.com.
[14] S. S. Haykin, “Neural Networks: A Comprehensive Foundation.” New York: Maxwell Macmillan International, 1994.
[15] J. Manslow, “Using a Neural Network in a Game: A Concrete Example.” In M. Deloura (Ed.), Game Programming Gems 2. Hingham, MA: Charles River Media, Inc., 2001, pp. 351-357.
[16] A. J. Champandard, “The Dark Art of Neural Networks.” In S. Rabin (Ed.), AI Game Programming Wisdom. Hingham, MA: Charles River Media, Inc., 2002, pp. 640-65.
[17] P. Keller, “Commercial Applications, Artificial Neural Networks.” Retrieved June 5, Pacific Northwestern Laboratory. 2002, from http://www.emsl.pnl.gov:2080/proj/neuron/neural/p roducts, 1997
[18] A. LaMothe, “A Neural-Net Primer.” In M. Deloura (Ed.), Game Programming Gems. Hingham, MA: Charles River Media, Inc, 2000, pp. 330-350.
[19] J .Manslow, “Imitating Random Variations in Behaviour Using a Neural Network.” In S. Rabin (Ed.), AI Game Programming Wisdom. Hingham,
MA: Charles River Media, Inc., 2002, pp. 624-628.
[20] S. Woodcock, “AI Roundtable Moderator’s Report.” Retrieved April 9, 2002, from http://www.gameai.com, 2002.
[21] M. Buckland, “Genetic Algorithms in Plain English.” Retrieved March 7, 2002, from http://www.btinternet.com/~fup/ga_tutorial.html.
[22] F. D. Laramee, “Genetic Algorithms: Evolving the Perfect Troll.” In S. Rabin (Ed.), AI Game Programming Wisdom. Hingham, MA: Charles River Media, Inc, 2002, pp. 629-639.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

[23] S. Hsiung,, J. Matthews, "An Introduction to Genetic Algorithm and Genetic Programming." Retrieved July 16, 2002, from http://www.generation5.org/ga.shtml, 2000.
[24] N. Dulay, "Application of Genetic Algorithm." Retrieved July 18, 2002, from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/tcw2/article1.html, 1996.
[25] N. Dulay, Genetic Algorithms. Retrieved July 18, 2002, from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html, 1996.
[26] G. James, "Using Genetic Algorithms for Game AI." Retrieved July 18, 2002, from http://www.gignews.com/gregjames1.htm, 2002
[27] N. Dimension, Inc. "Genetic Algorithms: Common Applications." Retrieved July 18, 2002, from http://www.nd.com/products/genetic/apps.htm, 2002.
[28] A. LaMothe, "Tricks of the Windows Game Programming Gurus." Indianapolis, Indiana: SAMS, 1999.
[29] Buckland, M. "Genetic Algorithms in Plain English." Retrieved March 7, 2002, from http://www.btinternet.com/~fup/ga_tutorial.html, 2002.
[30] M.Peter. "Interview with Gamespot" Online Magazine, March 2001 http://www.gamespot.com.
[31] "Game Programming Information", 5 May 2002, Videogiochi.net. http://www.videogiochi.net/cgi-bin/search/gamesearch.cgi?ID=938448137.
[32] F. John, "AI for Games and Animation" AK Peters, 1999.
[33] E. Richard, "Interview with divineorder.org", http://www.divineorder.org
[34] E. Richard. "The Future of AI in Games: A Personal View", Game Developer Magazine, August 2001.

Senzota Kivaria Semakuwa, Received the B.Sc. degrees in computer engineering and information Technology from The University of Dar es Salaam, Tanzania in 2010. He has published one conference paper with ISBN: 978-960-474-286-8. He is currently a Master student in the School of Information Science and Engineering, Central South University, Changsha China. He is working as a Tutorial assistant in the Department of Computing Science and studies at Mzumbe University, Tanzania. His research interests include multimedia database, ubiquitous computing, Artificial Intelligence, image processing and pattern recognition.

Florence Upendo Rashid, received B. Sc in Computer Science from Makerere University,Kampala,Uganda, ME in Communication and Information System from Wuhan,China.She is currently a PhD. Student in Communication and Information System at School of Information Science and Engineering, Hunan University, Changsha, Hunan Province – PR. China. She is currently a Assistant lecturer in the Department of Telecommunications and Computer Networks, School of Informatics,College of Informatics and Virtual Education,University Of Dodoma, Dodoma,Tanzania.She has authored one paper in Journal of Informatics and Virtual Education.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.