VOL. 18 NO. 9
ISSN: 2278-1897
SEPTEMBER, 2020

African Scholar Publications & Research International

African Scholars Journal of pure and Applied Science (JPAS-9)

Application of Non-Deterministic Finite Automata to Pro Evolution Soccer (PES) Game.

[1]Abubakar Muhammad Shafii; [2]Manko Abubakar Abdulrahman; [3]Ibrahim Aliyu; & [4]Odo, Chika Emelda

Department of Mathematics and Statistics, School of Basic and General Studies, the Federal Polytechnic, Bida

Abstract

In this research, Non-deterministic finite automata (NDFA) is used to design a finite state machine model for pro evolution soccer (PES) game, the game consists of a real-user who uses the game controller to play or control the activities of play. Inputs from the game controller is represented as a finite input symbol and the game character is represented as a set of finite states. Relationship between the sets of input symbols and the set of finite states is represented on a digraph using NDFA model which is used to define a transition function δ.

Keywords: Application, Non-Deterministic, Finite, Automata, Evolution.

Introduction

Finite state machines or finite automaton as the name implies is a branch of automata theory or computational theory that studies the automatic movements of machines which is caused by a certain change in the state of the machine, this concept has been used to model systems in diverse areas of science and technology. In a nutshell, we can define a finite state machine as a tool that is used to stimulate sequential logic, which is implemented where a system input can cause a certain change or changes in the state of a machine.

Example of a finite state machines includes the turnstile gate system which accepts inputs through the insertion of a coin and gives output by opening the gate, the gate returns to its initial state after the user has passed through it. Finite state machines (FSM) are divided into two namely: deterministic finite automata (DFA), and Non-deterministic finite automata (NDFA), we will be focusing on NDFAs.

Research in the study of finite state machines (FSM) has had variety of applications, among them is in game development which uses the idea of programming, other important fields is in automobile industries.

Mihalis and David (1991) defined finite state machines as a quintuple $M = \{I, O, S, \delta, \lambda\}$ where I,O and S are finite and nonempty sets of inputs symbols, output symbols and states respectively:

$\delta: S \times I \to S$, is the state of transition functions.

$\lambda: S \times I \to O$, is the output function.

Elmer and Soo (2012) presented a new algorithm for real time event using finite state machines with multiple fuzzy logic probability evaluators (FLPES), they further developed a machine referee for a robot soccer game which was used as the platform to test the proposed algorithm.

Furthermore, a novel technique to detect collision and other events in microbot soccer game under inaccurate and insufficient information was presented.

Past literature have shown how effectively the finite state machines have been used to model different games, softwares and machine learning, one of such literatures is the application of automata in game theory by Sally, Khaled, and Muhammad (2013) in which they argued that, in game theory, if players are presented in strategies, it directly affects the performance of the players, they also found that finite automata, adaptive automata, and cellular automata would be effective to this study, where the number of interactive players is high, and therefore, complex strategies are needed.

Ritika, Ahmed, and Qasim (2013) critically analyzed the success of the traditional reliability models built to measure and estimate software reliability. They further propose that a finite state automata based reliability model can serve as befitting solution to all existing software reliability challenges.

Pukeng et al. (2019) presented their study on game simulator-based education which they used to design and stimulate education games such as “flora the explorer, introduction to plant with android-based finite state machines”.

The swish max4 and shuffle algorithm was used to develop the game. The game agent uses finite state machine (FSM), this helps the game give notice to players when answers go wrong or right in the game.

Ada, Eka and Aldy (2019) applied finite state machines algorithm on 2D platformer Rabbit games Vs Zombies which was developed using android platform, the finite state machines (FSM) was used for the movement of Non-player character (NPC) to produce dynamic movements.

The Pro Evolution Soccer Game

The pro evolution soccer (PES) game is a game series that was developed by Konami in 2001, the game series has undergone annual update since then to keep up with the modernity of the gaming console "Play Station". The game series has sold over 106.8million copies and has been released on over ten platforms, among them are:

Play station, play station 2, play station 3, play station 4, xbox, android, IOS, windows and play station portable (PSP).

The game strives to emulate real soccer, as such , gameplay stimulates a typical game of association football, with the player controlling either an entire team or a selected player; objectives coincides with the rules of association of football. Various game modes have been featured in the series, allowing for gameplay variety, including the kick off, online and offline modes. Retrieved from https://en.m.wikipedia.org/wiki/pro_evolution_soccer

The Play Station Dual Shock 4 Controller

The play station 4 controller features a basic design of a D-Pad (Directional Pad), 4 main select buttons which includes (●■✖▲), two analog controls, the start and select buttons, the shoulder buttons and L_1, L_2, R_1, R_2 where L,R = left and right and 1,2 = top and bottom. Retrieved from https://en.m.wikipedia.org/wiki/playstation.

Definition 1.4 Let an NDFA be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where:

- Q is a finite set of states
- Σ is a finite set of input symbols
- δ is the transition function where $\delta: Q \times E \rightarrow 2^Q$
- q_0 is the initial state where $q_0 \boldsymbol{\epsilon Q}$
- F is a finite set of states where $Q(F \subseteq Q)$

Result and Discussion

In this research, our main focus is on the pro evolution soccer game, non-deterministic finite automata (NDFA) is used to design a finite state machine model for pro evolution soccer (PES) game, the game consists of a real-user who uses the game controller to play or control the activities of play. The following conditions must be put into consideration.

i. The game characters will be represented as a set of finite states.

ii. The buttons of the game controller will represent set of finite input symbols.

iii. A team shall consists of 11 game characters, the game characters shall consist of:

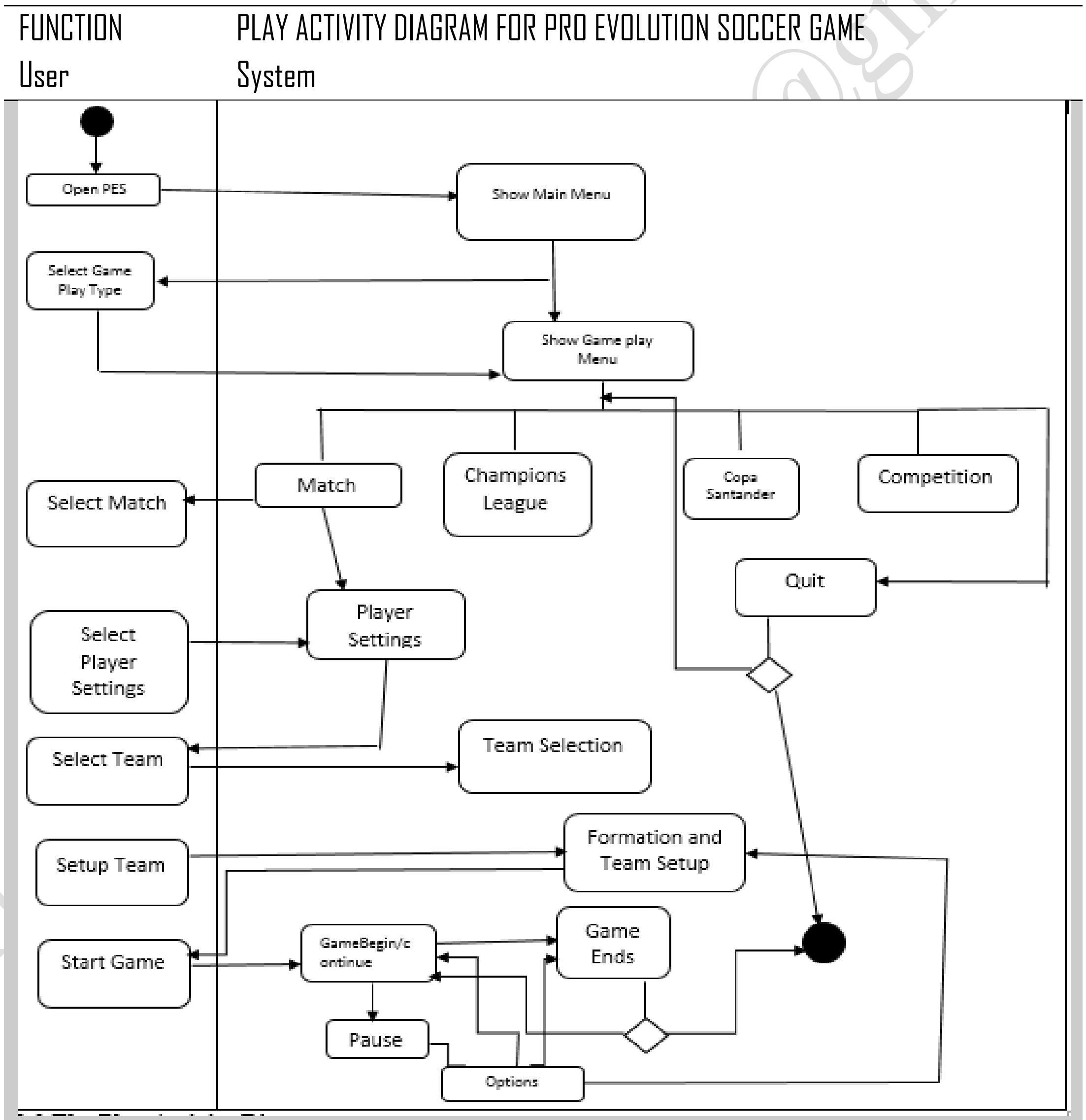
One goalkeeper,

Four defenders,
Four Midfielders,
Two Strikers. This will depend on the formational tactics adopted by a team.

iv. All game characters have the ability to run around the football pitch of play, which covers a total area of $7192.5m^2$ with length=105m and width=68.5m

v. The goalkeeper can freely hold the ball in his hands within 18yards of the penalty area.

Figure 2.2 Play Activity Diagram for starting the pro evolution soccer PES game



The Play Activity Diagram

Figure 2.2 illustrates the play activity diagram which shows how to start the game, play the game and end game after playing. The following explains the play activity diagram:

1. To start the game, *open PES,* this displays the **Main Menu**.
2. Select *Game Play Type,* this displays categories of game play, such as Match, Champions League, Copa Santander, Competition and the Quit option. If a user selects quit, he has options to either select "yes" or "no". The option 'no" returns a user to the **Game Play Menu** while option "yes" closes the game.
3. Select *player settings* to go to the player settings.
4. Select team to use for the gameplay, this option gives wide variety of team to select from.
5. Select *team setup*, this allows the setting of formation and player setup.
6. Finally, Start the game, during play, a user can **pause** the game to go-to other options such as formation and player setup. A user can also decide to continue the game or end the game, this gives options of "Yes" and "No", where "no" returns the user to continue the game and "yes" ends the game.

Table 2.3. The table below shows the meaning of buttons of the game controller and defines the input symbols.

Controller	Meaning	Input Symbol ($\boldsymbol{\Sigma}$)
▲	Through/Long Pass	1
■	Shoot	2
●	Cross	3
✖	Pass	4
Left Dpad	Move left	5
Right Dpad	Move right	6
Top Dpad	Move up	7
Bottom Dpad	Move down	8
$\boldsymbol{R_1}$	Run/Sprint	9
$\boldsymbol{R_2}$	Place ball	10
Left Analog	Directional analog	11

Table 2.4. Let A,…,k represent the states of the automata as shown in the below.

Game Character	States (Q)
Goalkeeper	A
Defender 1	B
Defender 2	C
Defender 3	D
Defender 4	E
Midfielder 1	F
Midfielder 2	G
Midfielder 3	H
Midfielder 4	I
Supporting Striker	J
Striker	K

From table 2.3 and table 2.4. Let Q be defined as a set of states defined by $Q = \{A, B, \ldots, K\}$ and $\Sigma = \{1,2,3,4,5,6,7,8,9,10,11\}$ be the sets of input symbols. We shall construct our digraph from table 2.3 and table 2.4.The

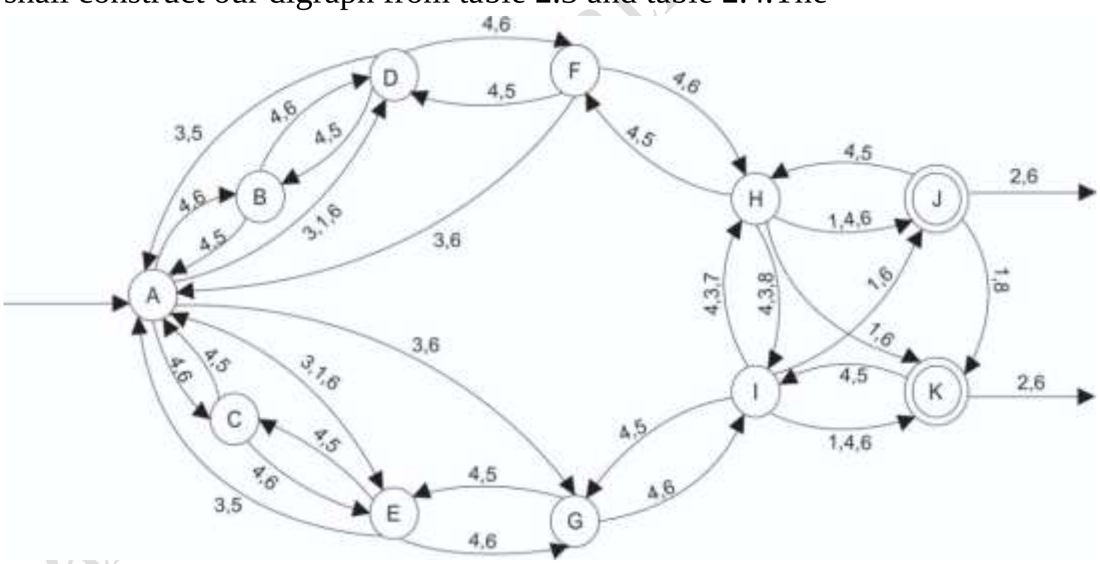


Figure 2.4. NDFA digraph Model that shows the possible behaviors of game characters on the pitch.

The NDFA system designed above describes the behavior of a game character when the is on possession. The behavior is determined by the user and it consists of two behaviors namely: the state and action of a game character. The NDFA model portrays 11 state, certain inputs like 5,6,7,8 can cause a change of state directionally and in some cases making the states overlap one another.

Certain inputs also define the actions of the states, actions available to a random state is pass, shoot, throw and foul. These actions are defined by inputs 1,2,3,4. If a wrong input is entered, it can cause the player's behavior to commit a fowl and every fowl will be an advantage to the defensive or opposing team.

Table 2.5. A Table of Transition Function for the NDFA digraph

	Next states for inputs										
Present State (Q)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
A	D,E		D,E,F,G	B,C		B,C,D,E,F,G					
B				D		D					
C				D		D					
D			A	F	A	F					
E			A	G	A	G					
F				H		H					
G				I		I					
H	J,K		I	I,J		J,K		I			
I	J		H	H,K	J	K	H				
J	K							K			
K											

From definition 1.4. Let $q_0 = A$, $Q = \{A, B, C, D, E, F, G, H, K\}$, $\Sigma = \{x: 1 \leq x \leq 11, x\epsilon N\}$, $\delta: Q \times \Sigma \to 2^Q$.

1. **State A**

   From the digraph, State A is the initial state, it has the ability to distribute short and long passes to states B and C, long passes to states D and E and crossing to midfield states F and G.

   State A is also an **ϵ**-state as it can hold a ball to its possession for some period of time before dispossession. The transition function is shown below:

   $A \times 1 \to \{D, E\}$

   $A \times 3 \to \{D, E$,F,G$\}$

   $A \times 4 \to \{B, C\}$

   $A \times 6 \to \{B, C, D, E, F, G\}$

2. **State B**

   State B has a limited ability according to the digraph, state B has the ability to give short passes to state D, long passes, shooting to score and crossing are undefined abilities of the defensive states, it is also an **ϵ**-state.

   The transition function is defined below:

   $B \times 4 \to \{D\}$

   $B \times 6 \to \{D\}$

3. **State C**
   The characteristics of state C is the same as that of state B including all undefined features.
   $C \times 4 \to \{D\}$
   $C \times 6 \to \{\}$
4. **State D**
   States D has ability to distribute short pass to state F, although it has an undefined characteristics to distribute long passes to the central midfield and crosses to the striking position. It also an **ϵ**-state
   The transition function is defined below:
   $D \times 3 \to \{A\}$
   $D \times 4 \to \{F\}$
   $D \times 5 \to \{A\}$
   $D \times 6 \to \{F\}$
5. **States E**
   The characteristics of state E is the same as that of state of D including all undefined features.
   $E \times 3 \to \{A\}$
   $E \times 4 \to \{G\}$
   $E \times 5 \to \{A\}$
   $E \times 6 \to \{G\}$
6. **State F**
   The midfield state F has the ability to pass the ball state H, it has an undefined ability to send a through pass to the striking positions and it is also an **ϵ**-state. The transition function is:
   $F \times 4 \to \{H\}$
   $F \times 6 \to \{H\}$
7. **State G**
   The characteristics of state G is the same as that of state F including all undefined features.
   The transition function is defined as:
   $G \times 4 \to \{I\}$
   $G \times 6 \to \{I\}$
8. **State H**
   This state plays a very important role such as supplying short passes to state J, through passes to states K, and crosses to state I. it can score when on target, it is also an **ϵ**-state and it can also decide to release the ball backwards to the midfields and defense.
   The transition function is defined below:
   $H \times 1 \to \{J, K\}$
   $H \times 3 \to \{I\}$

$H \times 4 \rightarrow \{I, J\}$
$H \times 6 \rightarrow \{J, K\}$
$H \times 8 \rightarrow \{I\}$

9. **State I**

This is another important state, it supplies passes to K, long passes to J and crosses to H, it can also decide to release the ball backwards to the midfields and defense.

The transition function is defined below:

$I \times 1 \rightarrow \{J\}$
$I \times 3 \rightarrow \{H\}$
$I \times 4 \rightarrow \{H, K\}$
$I \times 5 \rightarrow \{J\}$
$I \times 6 \rightarrow \{K\}$
$I \times 7 \rightarrow \{H\}$

10. **State J**

This is a final state, it is also known as a supporting strike state that can score a goal if on target, the role of this state is to produce goals and also assist state K with short and through passes.

It can return the ball to the midfield in case of defensive attack, it is also an **ϵ**-state.

The transition function is defined below:

$J \times 1 \rightarrow \{K\}$
$J \times 8 \rightarrow \{K\}$

11. **State K**

This is another final state whose duty is to produce goals, it can distribute passes to the supporting striker and close by midfielders although that is not define. It is also an **ϵ**-state.

The transition function for this state is undefined as it has not connectivity defined on any state on the digraph except to carry out actions of shooting the ball against the defensive teams.

Proposition 2.6. If $x \in X$ and X⊆Q, $n \in \Sigma \forall$n ≥ 1 then $\exists \delta : x \times n \rightarrow 2^X$ for every $X \geq 0$.

Proof: Let $x \in X$, and let $n \in \Sigma$, then the action $x \times n \rightarrow 2^X$ exists ∀ |X|=n, and the number of subsets of X is $|P(X)| = 2^X$. QED.

Proposition 2.7. In an NDFA digraph, the sum of degrees of the states is equal to twice the number of inputs.

Proof: Let $Q = \sum$xϵX deg (x) for $X \subseteq Q$ and $\Sigma = k_n \forall 1 \leq i \leq n$, in counting Q, we count each input twice. Thus, Q=2$|k_n|$ (The sum of degrees is twice the number of edges). Since S is even, then the number of states with odd degree is even.

Conclusion

At the end of this research, it is evident that, in pro evolution soccer game, every state has the ability to move in all directions, all players exhibit an ∈-state, this is a state that allows states to hold the ball for some period of time for the purpose of skills. Also, the number of input symbols that a state can act upon at a given time depends on the characteristics of the state.

The definition of NDFA holds for all input symbols on a state, this is shown in proposition 2.6.

It is also concluded that certain wrong input on a state at a particular time can cause a fowl, for example when a state loses possession of a ball and tries to cross the ball with input 3, this will give rise to a reckless tackle. All states can sprint and place ball by pressing input 9 and 10. Sprinting allows a state to run faster and placing of a ball gives room for clean goal with directions.

We can also see that, there is a walk connecting certain states, and the walks have repeated edges, therefore, we can say that, there is a close walk between states A and D, A and E, A and B, A and C, C and E, B and D, D and F, E and G, F and H, G and I, H and I, H and J, I and k and J and K. According to the NDFA, the trails are, A and F, A and G, H and K, J and I, and J and K.

References

Ade, S. Eka, W.H. and Aldy, P.A. (2019). Application of the Finite State Machine Algorithm on 2D Platformer Rabbit Games vs Zombies. *Jurnal Online Informatika*, Vol.4, No.1,33-38.

Dominique C. and Richard E.S. (2018). A Thermodynamically Consistent Model of Finite State Machines. *The Royal Society Publishing*. **8:**20180037

Elmer P.D. and Soo H.P. (2009). Real Time Robot Soccer Game Event Detection Using Finite State Machines With Multiple Fuzzy Logic Probability Evaluators. *International Journal of Computer Games Technology*. Vol. 2009, Article ID 375905, 12 pages.

Mihalis Y. and David L. (1991). Testing Finite State Machines: Fault Detection. *Journal of Computer and System Sciences*, **50,** 209-227.

Pukeng A.F, Fauzi R.R, Lilyana, Andrea A, Yulsilviana E and Mallal S. (2019). An Intelligent Agent of Finite State Machine in Educational Game "Flora the Explorer". *Journal of Physics*, **1341**(2019)042006.

Ritika W, Ahmed P, and Qasim M.R. (2013). Automata Based Software Relaibility Model: The Key to Reliable Software. *International Journal of Sowftware Engineering and its Applications*, vol.7,No.6(20143).pp.111-126.

Sally A, Khaled S, and Muhammed R. (2013). The Applications of Automata Theory in Game Theory. *Information Science Reference*.

Swester P, and Wiles, J. (2002). Current AI in Games: A review. *Australian Journal of Intelligent Information Processing Systems*, 8(1), pp. 24-42.

https://en.m.wikipedia.org/wiki/pro_evolution_soccer

https://en.m.wikipedia.org/wiki/playstation