Computing Game Design with Automata Theory

Noman Sohaib Qureshi[1], Hassan Mushtaq[2], Muhammad Shehzad Aslam[2], Muhammad Ahsan[2], Mohsin Ali[2] and Muhammad Aqib Atta[2]

Abstract– **Use of Computational and Automata Theory is common now a day in lexical analysis in compilers and programming languages, morphological analysis, image compression, bioinformatics etc. However in computer games and computing game theory its use is not widespread. This paper summarizes the design of an arcade game using automata theory tools. Deterministic finite state automaton (DFSA) and non-deterministic finite state automaton (NDFSA) are used primarily in various levels of designing the game.**

Keywords– **Morphological analysis, Image Compression, DFSA and NDFSA**

I. INTRODUCTION

Alan Turing, the great scientist, and father of computing has contributed to the birth of automata theory and the biggest contribution *'Turing Machines'* has almost solved David Hilbert's famous 10^{th} problem. Though usage of these computational tools is common at macro level research areas however in disciplines like computer video games, it is in its inception. And there is room for research in this area. Computational and automata tools may play a significant advancement role in the design and development of computer games and computing game theory.

In this paper an effort is done to promote usage of these tools and computational theory in game design. The design of arcade game presented in this paper contains five levels. At each level the player is encountered with various states and enemies. Few default weapons are deployed to the player and at certain states depending on the input alphabet player may get various other weapons. At each level from level one to five, the player first has to defeat minor enemies and then leading towards challenging game stages with major enemies. Input Alphabet $\sum$ is stated before all the levels and for each level a macro automaton is designed which is further split into detailed automata. Certain regular expressions are designed to be fed as input strings for the game progress by player.

At each level certain states from set Q in the automata are designated for point gains or losses by the player. Also, the weapons and ammo is distributed at various stages in game by reading or reaching at a state from set Q. Input alphabet $\sum$ of the designed automata is divided into weapon, select, move, action and game sets. States are assigned labels or tags and are explained. The state routing in game is smooth at from level one to level five and it is designed and constructed using DFSA and NDFSA tools.

[1]Lecturer, Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), Pakistan

[2]Research Scholars (Automata and Computational Theory Group), Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), Pakistan

II. EXISTING WORK REVIEW

In 1950, John Nash [1] demonstrated that finite games have always have an equilibrium point, at which all players choose actions which are best for them given their opponents choices. This central concept of non-cooperative game theory has been a focal point of analysis since then. In the 1950s and 1960s, game theory was expanded theoretically and applied to other problems such as war and politics.

As a mathematical tool for the decision-maker the strength of game theory is the methodology which provides structuring and analyzing problems. The formal modeling of a situation as a game requires the decision-maker to enumerate explicitly the players and their strategic options, and to consider their preferences. So game theory plays a vital role in game designing [2].

Clock games are other games using a clock in which player has to perform some actions in specific duration. These things can be handled by using the tools of game theory. Players in clock games must decide the timing of some strategic action. First, unlike the theory model, it is found that a longer time duration before players received the signal that the time was "ripe" led to shorter delay and, in some cases, no delay at all. The force of the "calendar" effect differed depending on the speed of information diffusion. Finally, the possibility of mistakes, especially when moves were observable, led to less herding behavior than was predicted by the theory [3].

In game theory usually analyses is performed regarding decision-making processes in various fields. There are different methods of solving decision related problems. Game theory focuses on one end on the problem solution from one player's point of view, whereas on the other end it also solves the problems in the interaction among many players. Much of the game theory is concerned with finite, discrete games which have a finite number of players, moves, events and outcomes etc. Many researchers from multidisciplinary fields have worked and applied the game theory in construction engineering and management domains. Game theory is an appropriate approach for decision making in construction engineering and management processes to solving different problem from real life [4].

The use of educational games in learning environments is an increasing trend. This trade can play a very important role in the quality of the educational experience, allowing the learning environment for students with different learning styles, different levels of initial knowledge and different expectations and objectives. These games broad the vision of students. Videogames are inherently interactive and reactive to the actions of the user and they are complex pieces of software being executed in the student's computer, which facilitates the student to learn more quickly and to memorize. Games are a very rich interactive medium, and this interactive behavior can be exploited for assessment purposes**.** Preliminary students get more advantage from these games. These games familiarize the students with real world [5].

In traditional computer games, the player has a double role as both observer and an actor. Pervasive gaming goes even further in complicating the coupling of identity and structure, as these games reflect directly the player's reality and construct a second world [6].

The game rules can be converted to fitness functions. Fitness functions are used to determine how to parameterize game. For constructing fitness functions from theories, these are divided into static and dynamic theories. Static theories are those which do not require the player to learn while playing. By which one can judge personal qualities of the player at any moment. In contrast, dynamic theories put learning in focus, and the entertainment of a game could only be judged by how the agent or player adapts to the game over time [7].

The building of computational caricatures is a sharing opinionated claim about game design and the technology that should power future design automation systems. The game design with artificial intelligence tells us about human game design and future design assisted by and embedded in machines. Computational caricature appears to be more effective at exploring deeply technical ideas about the design process [8].

III. THE DESIGNED GAME WITH COMPUTATIONAL THEORY

There are good and danger zone in every level. Player is allowed to choose one of these zones. In good zone there are fewer enemies giving more points to player. But in danger zone there are more enemies relatively giving fewer points to player. In both the zones of each level some enemies will be hidden and player cannot safely pass from there, if these enemies hit player, he will lose more power. Hidden enemies give more points or some weapon to player. Player can choose different path to reach to a specific enemy (not for all enemies).

A. Game Description

There are total 5 levels in Game starting from level 1 to level 5 as elaborated in the following section.

1) Level 1

Player is Pedestal having knife, pistol and conditional weapons.

Enemies:

- 20 pedestal soldiers (each power 100) having pistols
- At the end of level, one big enemy (power of 10 pedestals i.e. 1000) "Zion" is present having AK-47

2) Level 2

Player is provided a Machine gun van as default weapon.

Enemies:

- 25 pedestal soldiers (each power 100) having Uzi
- 12 soldiers on bikes (each power 100) having AK-47
- At end of level two big enemies (each power of 5 biker i.e. 500) "Helicopter" having cannon is present

3) Level 3

Player is provided a Tank capable of firing cannon and machine gun. He is provided with 35 cannons and 500 rounds of machine gun to complete level 3.

Enemies:

- 12 pedestal soldiers (each power 100) having Uzi
- 3 machine gunner (each power 100)
- 10 Tank (each power 100) having cannon
- At end of level three, big enemy (power of 5 tanks i.e. 500) "Planet machine" flying and having 'cannon' and machine gun are deployed

4) Level 4

Player is provided a Helicopter capable of firing cannon and machine gun. Also, he is provided with 18 cannons and 300 rounds of machine gun and a sniper with 20 rounds.

Enemies:

- 5 Machine gunner (each power 100)
- 5 Helicopter (each power 100) having cannon
- 5 Tank (each power 100) having cannon
- At end of level 1 big enemy (power of 10 helicopters i.e. 1000) "Zous" having cannon and sniper

5) Level 5

Player is pedestal first having AK-47 with 30 bullets. He has to clear sector S-1 in order to occupy an F-16 of enemies that is capable of firing guided missiles. He is provided with 25 missiles to complete level 5.

Enemies:

- 10 pedestal soldiers (each power 100) having AK-47
- 5 jet plane (each power 100) having missiles
- At end of level five, a big enemy (power of 15 jet planes i.e. 15000) "Morpheus", a flying machine having cannon and missiles is deployed

Power description of player:

- Initial power: 200 (2 lives)
- Death power: less than or equal to zero
- Initial score: points zero
- 600 score points gives one life (i.e. 100 power)

Power loss of player:

1. Pedestal: 1 pistol bullet = -20, 1 AK-47 bullet= -30
2. Van: 1 Uzi bullet = -5, 1 AK-47 bullet= -10, 1 cannon= -75
3. Tank: 1 Uzi bullet= -2, 1 machine gun bullet= -4, 1 cannon= -25

4. Helicopter: 1 machine gun bullet= -4, 1 Sniper bullet= -10, 1 cannon= -25
5. F-16: 1 cannon= -10, 1 missile= -20

Power loss of enemies:

1. Pedestal soldier: 1 hit of knife= -100, 1 pistol bullet= -50, any other weapon bullet= -100.
2. Zion: 1 pistol bullet= -50, machine gun bullet= -100, hand grenade= -100.
3. Biker: 1 round of machine gun= -50.
4. Helicopter: 1 cannon= -50, 1 sniper bullet= -25, 1 machine gun bullet= -10.
5. Tank: 1 cannon= -50.
6. Planet machine: 1 cannon= -50, 1 machine gun bullet= -5.
7. Zous: 1 cannon= -50, 1 sniper bullet= -25, 1 machine gun bullet= -10.
8. Jet plane: 1 missile= -100.
9. Morpheus: 1 missile= -100.

B. Defining above game in terms of Automata theory

1) Defining input alphabet:

Σ_{Weapon}= {K, P, A, H, M, S, C, U, Mi}
Σ_{Select}= {g, d}
Σ_{Move}= {25, 50, 75, 100}
Σ_{Action}= {191019, 102010, 701070}
Σ_{Game}= {go, gc, gp, eo, Ammo, -Ammo, gz, gd, -gd, gg, -gg}

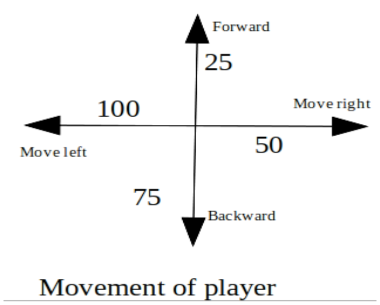

Fig. 1 : Movement of the Player

2) Description of input alphabet:

1. Σ_{Weapon}: It defines weapons that are provided to player or enemy during a mission. K (knife), P (pistol), A (AK-47), H (hand garnade), M (machine gun), S (Sniper), C (cannon), U (Uzi) and Mi (missiles)
2. Σ_{Select} allow the player to choose a zone g (good zone less enemy) and d (danger zone more enemy).
3. Σ_{Move} it defines movement of player 25 (forward), 50 (move right), 75 (backward), 100 (move left). It is shown in Fig. 1.
4. Σ_{Action} it defines an action 191019 (player hit enemy), 102010 (enemy hit player), 701070 (player save from enemy hit).
5. Σ_{Game} it defines some conditions of game go (player power<= 0), gc (player has points more than some specified points: (it will use to check that player is in position to fight with big enemy)), gp (player point less than specified points), eo (enemy died), Ammo (player has specific type of Ammo), -Ammo (player does not have specific type of Ammo), gz (checks game zone of specific level), gd (danger zone of specific level not completed), -gd (danger zone of specific level completed, gg (good zone of specific level not completed), -gg (good zone of specific level completed).

3) Some important states of Game:

Start state (q_0): Start Game.
Win state (F): End Game.
Loose state: Exit game.
Q_{source}= {Wep, source M-Van, source Tank, source Heli, source F-16}

State	Description
Exit game	Game exit depending on "go" or "gp".
Wep	Player is given a specific type of weapon.
Source	Player is provided with a source.
Pt	Player is given some points.
Pow -	Player power deducted.
Pow	Player has same power and points.
Pe	Power of enemy is reduced.
Stay	Player is roaring around not going to next enemy
Fire	Player fired.
Enemy fire	Enemy's fire hit the players.

Table 1: State Description of Designed Game

The player will get some power point on killing an enemy. It will get extra points by killing specific enemies. Hidden enemies will give more power point than other. It can go to either good zone or danger zone. If it selects a zone, it cannot change its zone (usually). At the end of each level and zone player will get some points and at the end of each level it has to kill a big enemy. For going to kill a big enemy player must get some specific points otherwise "gp" will occur taking player to lose state.

There are three possibilities of fight:

1. Player hits enemy
2. Enemy hits player
3. Player saves and move away leaving enemy (but can't save in case of hidden or specific enemies).

If enemy hits the player then player losses some of its power depending on the situation (i.e. type of fire that hit the player) and if it does not move back, left or right he continuously is hit by the enemy and which results in losing its power and if the power becomes zero (go) than it will die (i.e., game exit). Player can use different weapons that are provided to him for killing the enemies; however it must possess that type of ammo. A macro automaton of all the levels in the game is shown in Fig. 2. This automaton is expanded for each level to accommodate certain states.

goes on to fight with "Zion". If Zion is finished then level 1 is cleared, and the player is promoted to level 2. A macro automaton of all the levels in the game is shown in Fig. 2. This automaton is expanded for each level to accommodate certain states.

Player can use K, or P to kill enemies until he get some other weapon. Player will kill enemy with knife if regular expression K191019 occurs. It will kill enemy with pistol if R.E. $P(191019)^2$ will occur. Player hit by enemy if R.E.

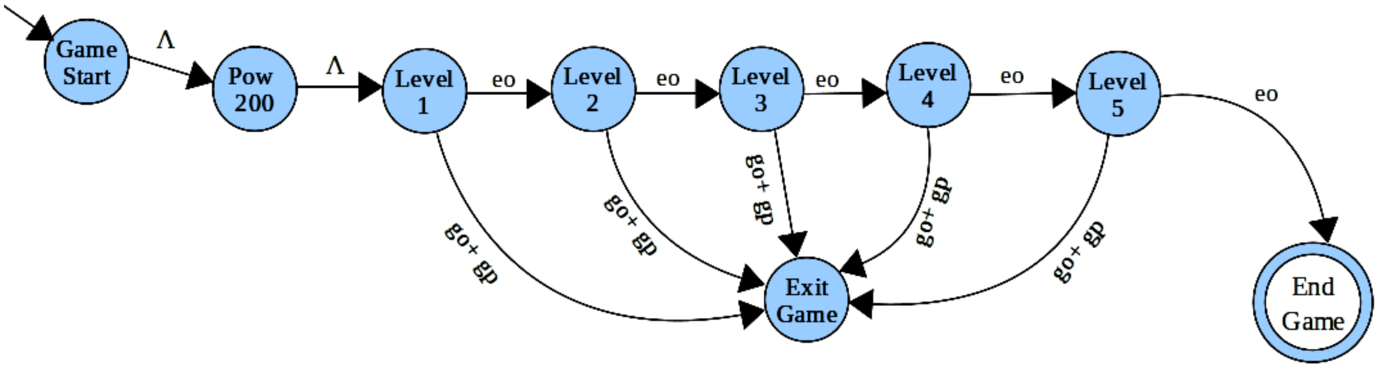

Fig. 2: Marco Level Automaton of Overall Game Design

4) Start Game:
Player is provided power of 200 points. It is sent to level 1.

a) Level 1

A map of level 1 is shown is Fig. 3. The automaton is further expanded to automata presented in Fig. 3.1 to Fig. 3.4. Player is provided with weapon K and P with as many rounds

(K+P)102010 will occur. Player save if R.E. (P+K)701070 will occur. He has to go forward (25) if he want to go to next enemy, otherwise he will be roaring around. Than if he able to get 'A' he can kill enemies if R.E. A191019 will occur, and so on with other weapons he found. There are different weapons he gets so different possibilities to go to D-Z (danger zone).

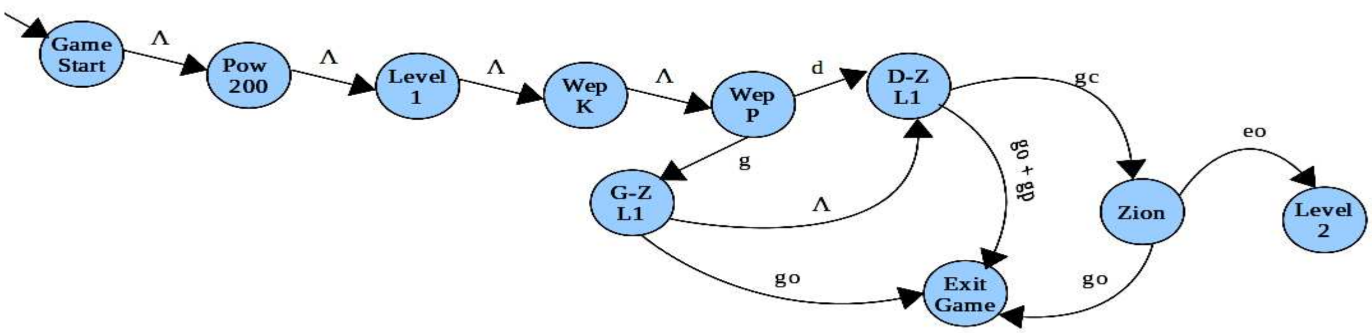

Fig. 3: Showing Level 1

As he can use. These are the default weapons in this level. He must gain 400 points to kill "Zion", i.e. for "gc" to occur; otherwise "gp" will occur. Player selects 'g' or'd'. If he selects d then he has to kill a single enemy. Otherwise if he wants to go back to status (75) he will go to good zone, and gets 20 points. If he does not want to go back then he will die because there are many enemies and game will exit. If he selects 'g' then playing game will result in killing enemies and the player will get 50 points on each. If enemy hits player it losses power equal to 20 points continuously until it move forward. If "go" state is found, then the game will exit. After clearing good and danger zone if "gp" is encountered, then the game will exit and if "gc" is found on the way then the player

He will go to 'Zion' with different weapons. So for each possibility NFA is presented in Fig. 3.1 to Fig. 3.4. The epsilon edges consumed no input alphabet from ∑ but this route is travelled without any input string fed into the process of game progress. Rest of the edges contains certain input strings to go to certain state from Q in the non-deterministic finite state automata presented. The beauty of using computational and automata theory in game design lies in the fact that it simplifies so much the process of game design and automation with rich scenarios and more possible moves with ease as is shown in the NFAs for this arcade game with less effort in design and more robustness and efficiency is obtained.

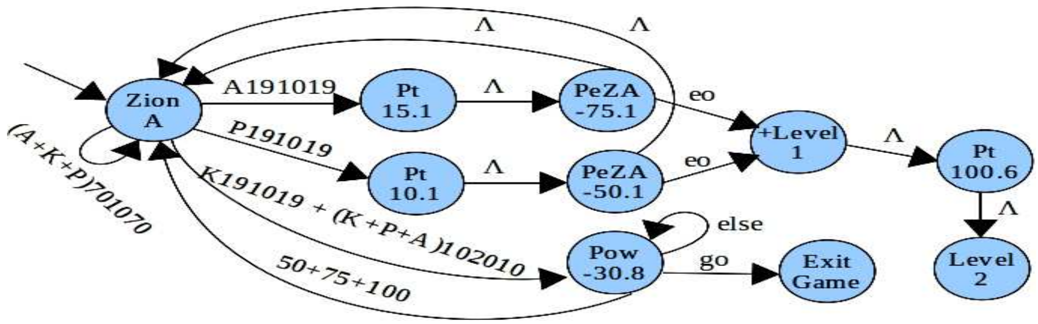

Fig. 3.1: Level 1 Zion A Zone

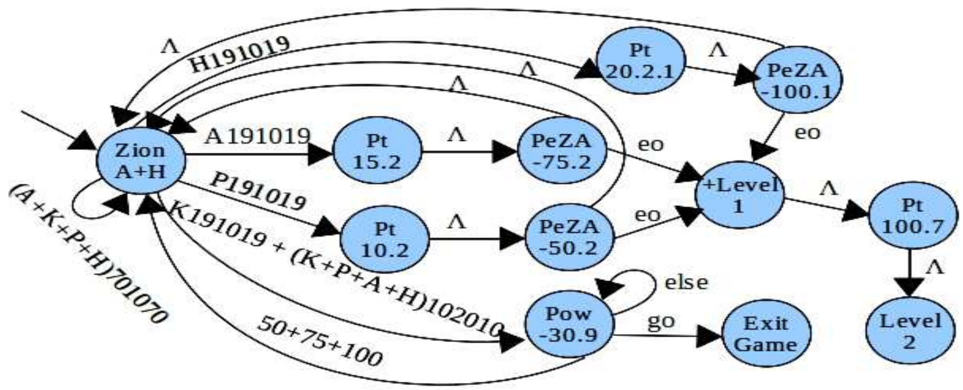

Fig. 3.2: Level 1 Zion (A + H) Zone

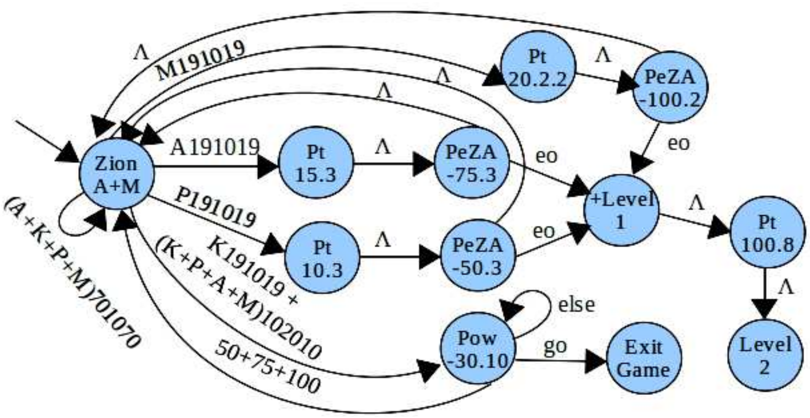

Fig. 3.3: Level 1 Zion (A + M) Zone

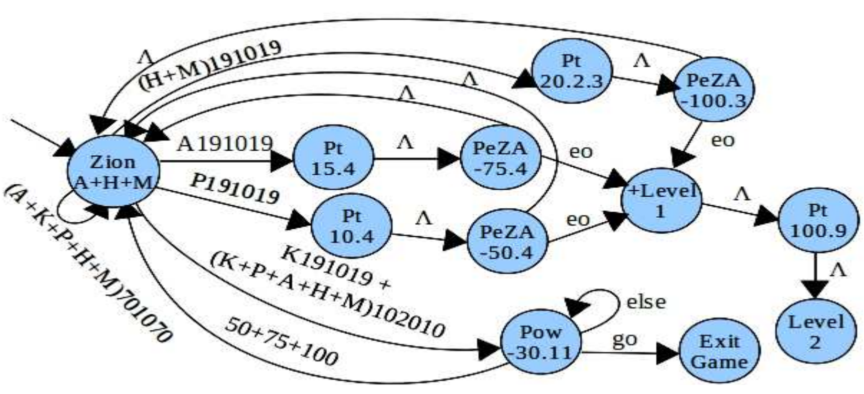

Fig. 3.4: Level 1 Zion (A+H+M) Zone

b) Level 2

Level 2 automatons are presented in Fig. 4 which is further expanded in automaton presented in Fig. 4.1. Player is provided with a machine gun van with as many rounds as he can use. He can go to either good or danger zone. He can kill enemies by R.E. M191019 or also can save. There are different enemies with different weapons. So power loss will be different if the enemy hits. For killing big enemy he must score 900 points.

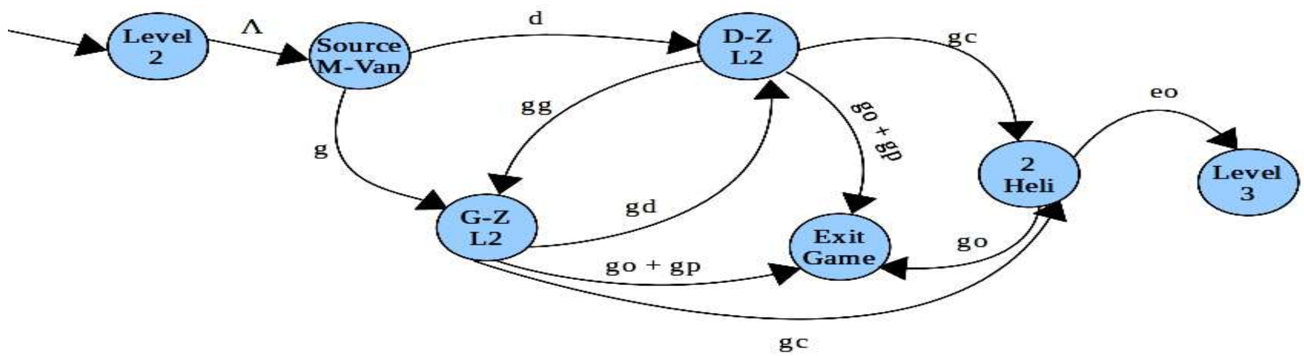

Fig. 4: Level 2 Automaton

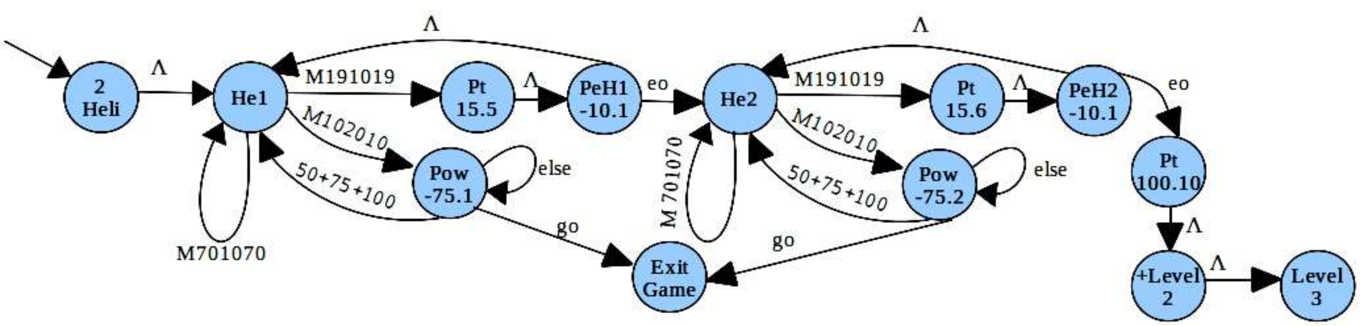

Fig. 4.1: Level 2 Heli Zone

c) Level 3

A map of level 3 is shown is Fig. 5 which is further expanded in Fig. 5.1. Player is provided with tank. He can go to either good or danger zone, and can use weapon 'M' or 'C'. Enemy tanks can only be destroyed by weapon 'C'. Player must score 1500 points to go to planet machine. If "eo" found, he is promoted to level 4. If "go" found at any stage during this scenario of play, the game will exit.

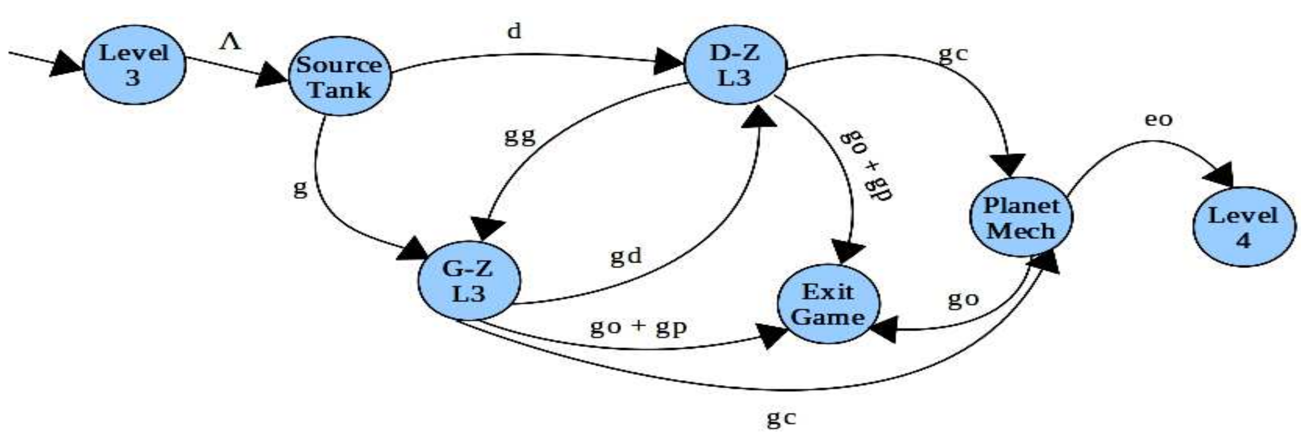

Fig. 5: Level 3 Automaton

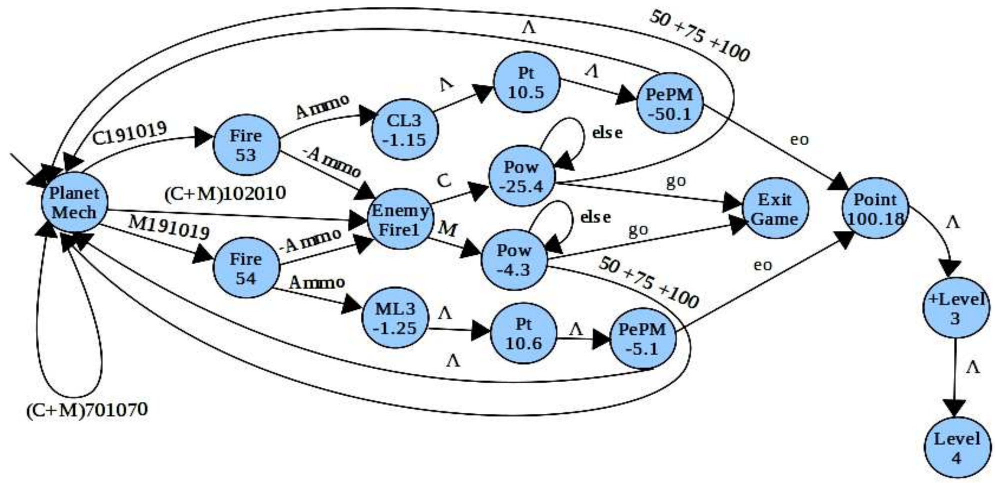

Fig. 5.1: Level 3 Planet Mech Zone

d) Level 4

Automaton of level 4 is shown in Fig. 6 which is further expanded in Fig. 6.1. Player is provided with a helicopter with weapon 'C', 'M' or 'S'. Enemy tank can only be destroyed by 'C' whereas helicopters can be destroyed by any of the weapons provided. Player chooses a zone, after clearing both zones. Points are checked and if "gp" is found in the way then the game will exit. Otherwise, the player is promoted to "Zous", and "eo" player is promoted to level 5. At any stage, if "go" is encountered in the path, the game will exit. Also, if "gp" is found just before promoting to "Zous" then again the game will exit. Player must score 2200 points to go to "Zous".

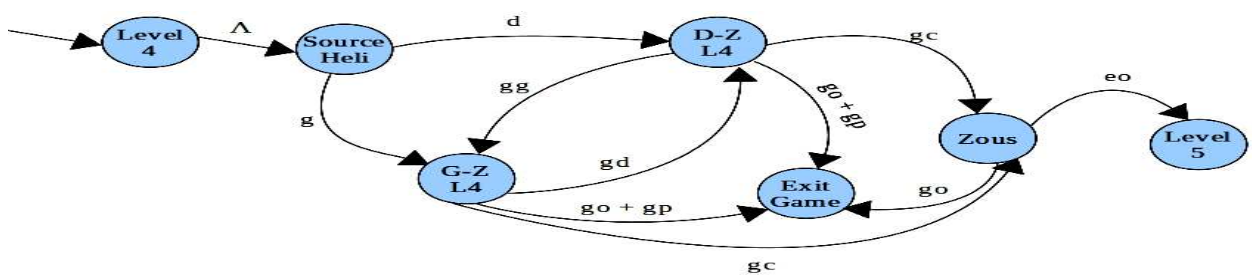

Fig. 6: Level 4 Automaton

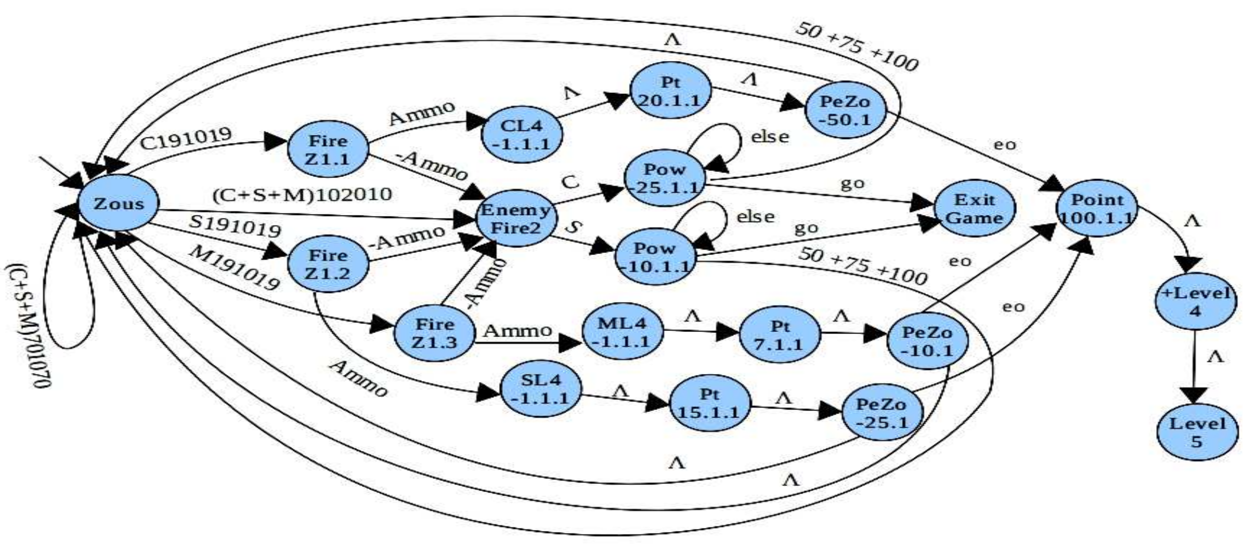

Fig. 6.1: Level 4 Zous Zone

e) Level 5

A macro level automaton of level 5 is presented is Fig. 7 which is further expanded in Fig. 7.1 to Fig. 7.2. Player is provided with weapon A. It must clear S-1 for getting control of an F-16. After clearing S-1 stage, it will be provided with F-16 and promoted to S-2, where he has to destroy 'Jet planes'. Then after clearing S-1, if "gp" is the entered as input alphabet string then the game will exit. Otherwise the player is promoted to "Morpheus". If "eo" is the input string then the game is completed. Also, at any time if "go" input is read from ∑ then game will exit. The player must score 2700 points to go to "Morpheus". By killing 'Morpheus' player will win (i.e. win state).

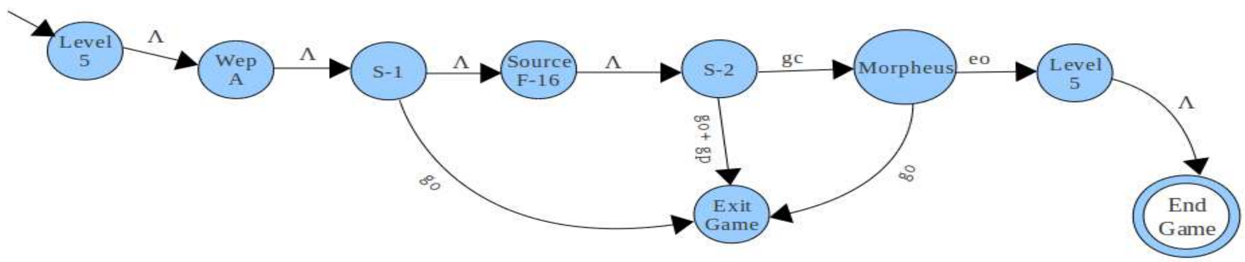

Fig. 7: Level 5 Automaton

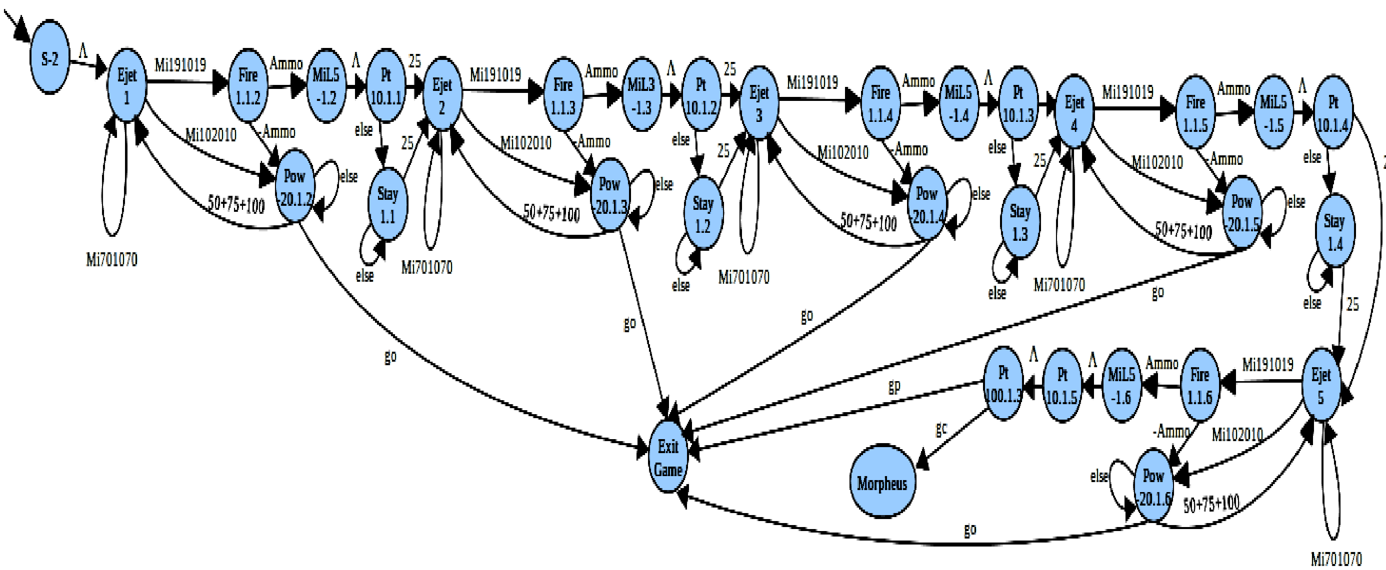

Fig. 7.1: Level 5 States

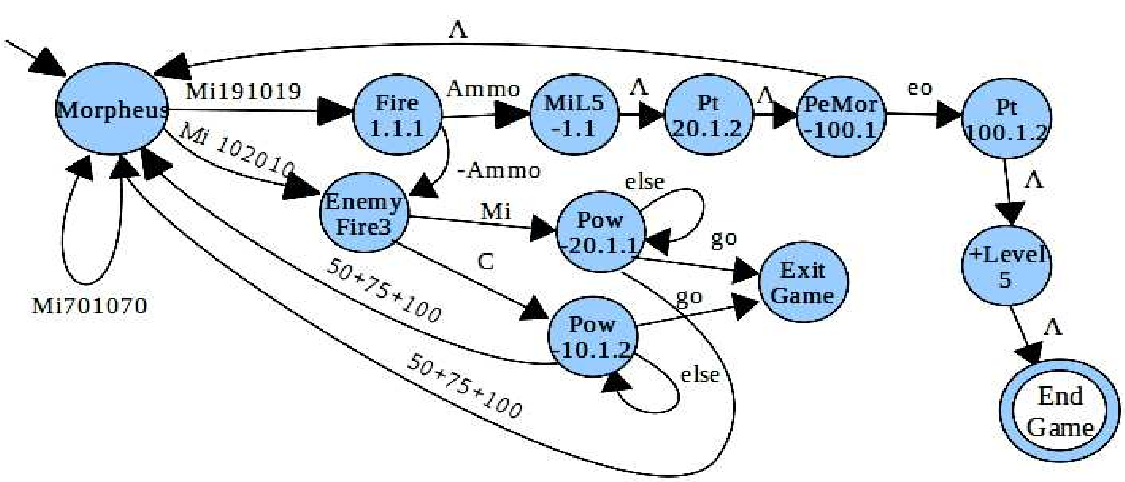

Fig. 7.2: Level 5 Morpheus Zone

IV. CONCLUSION

The initial design of arcade game is presented in this paper. The research group is working on the project of coding and development of this arcade game using automata and computational theory tools as presented. This research in future may further be extended to include advanced computational and complexity theory tools to further enhance the game theory to advanced artificial intelligence level using *'Turing Machines', 'Push Down Automata', 'Transducers and Weighted Transducers'* and *'Moore and Mealy Machines'*. The research regarding usage of automata theory tools in game design may further be extended to design and program real life scenarios, e.g. disease diagnostics, inventory management, business and strategic planning. The difference will only be the input alphabet ∑ and state scenarios. The basic theme of applying automata theory to real world problems will remain intact as just applied to game theory. The limitation of automata theory tools to mapping in computing games is that non-deterministic finite state automata cannot be directly programmed into a language until it is converted back to deterministic finite state automata. Because in programming languages the epsilon edges in non-deterministic automata cannot be translated to commands. As computer is a deterministic machine which is instructed to perform a task but at non-determinism level it's the human brain that works better. To translate the non-deterministic finite state automaton (NDFSA) logic unification and minimization may be applied using Kleene's theorem [9] of unification to obtain the regular expression which may be translated then to deterministic finite state automaton which is ideal to be programmed in any computing language. Despite this limitation *'applied automata theory'* is a discipline, in its inception of course in applied nature, in which applied research may result advancement in many fields not only limited to computing but also to social and medical sciences.

REFERENCES

[1] J. C. H. R. S. J. W. W. E. v. D. J. F. N. P. H. Harold W. Kuhn, "The Work of John Nash in Game Theory," in *journal of economic theory*, 1994.

[2] T. L. B. v. S. Turocy, "Game Theory," Encyclopedia of Information Systems, Academic Press, 2002.

[3] J. M. Markus K. Brunnermeier, "Clock Games: Theory and Experiments," manuscript, Haas School, UC Berkeley, 2008.

[4] J. T. Oleg Kapliński, "Game Theory Applications in Construction Engineering and Management," *Baltic Journal on Sustainability,* vol. 16, no. 2, p. 348–363, 2010.

[5] D. B. I. M.-O. J. L. S. B. F.-M. Pablo Moreno-Ger, "Educational game design for online education," *Computers in Human Behavior,* vol. 24, no. 6, pp. 2530-2540, 2008.

[6] B. K. Walther, "Pervasive Game-Play: Theoretical Reflections And Classifications," Center for Media Studies, University of Southern Denmark,, Denmark, 2006.

[7] J. S. Julian Togelius, "An Experiment in Automatic Game Design," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 2008.

[8] M. M. Adam M. Smith, "Computational Caricatures: Probing the Game Design Process with AI," Expressive Intelligence Studio, University of California, Santa Cruz, 2011.

[9] D. I. A. Cohen, Introduction to Computer Theory Automata Theory, First ed., John Wiley & Sons, Inc., 1986, pp. 100-138.