

Snakes and Stairs Game Design using Automata Theory

N. Raj^{1*}, R. Dubey²

¹Dept. of CSE, Lakshmi Narain College of Technology (RGPV University), Bhopal, India

²Dept. of CSE, Lakshmi Narain College of Technology (RGPV University), Bhopal, India

Corresponding Author: shivamedward@gmail.com

Online Available at: www.ijcseonline.org

Received: 23/Apr/2017, Revised: 02/May/2017, Accepted: 21/May/2017, Published: 30/May/2017

Abstract— Game design using automata tools and game theory analysis has been works from decades but there are a very few works on dynamic input or dynamic state generation. This paper summarizes game design of a traditional Snake and Ladder board game in a modernized way using the study of automata theory and game theory too, for both android and computer platforms with generation of random inputs which are also explained using flowcharts, DFSA and NDFSA.

Keywords—Automata Theory; Game Theory; Dynamic Game; Android

I. INTRODUCTION

Automata are so easy to understand that when they are integrated with the Game Theory there will be a great advancement in the field of Computer Game design. Not only computer games but also android games (which is popular in this tech era) or let's say any game. Computational and automata tools may play a significant role in the design and development of game theory.

This paper explains a simple game that is designed using the basic automata study and knowledge which includes DFSA and NDFSA, Regular Expressions, Algorithm and flowchart (for the code of game on android and computer system) related to them and the game flowchart and algorithm for easiness of user to understand the working of the game.

Everyone must have played the Snakes and Ladders board game. Maximum of four players can play the game and minimum are two. They are provided with a dice (set of 6 inputs Σ) which will be rolled by any one among them (set of players Q) and if he got a 6 on the dice then he will start the game (change the states Q) but if he doesn't then the other player will get a chance to do the same. As the game consists of snakes and ladders the player will have to pass the obstacles and the first one to reach the WIN state will be declared as the winner.

The players will get a chance to choose the difficulty level (set of difficulties Σ) among easy, medium, hard. Automata is designed for the easy and medium levels for the input alphabets from the dice.

II. RESEARCH SURVEY

Game theory is the formal study of conflict and cooperation. Game theoretic concepts apply whenever the actions of several agents (or players) are interdependent. These agents may be individuals, groups, firms, or any combination of these. The concepts of game theory provides a language to formulate, structure, analyze and understand strategic scenarios. Game theory is the study of human behavior in strategic settings. It is used to solve some harder problems in economics.

For every game, according to Game Theory there are different Solution Concepts. The most basic solution concept is Dominant Strategy Equilibrium, and this equilibrium occurs when the players play a strictly dominant strategy towards other. Suppose in a game there are two players and both are playing with their best strategies. When both the players play with the best strategies to each other, the equilibrium formed there is called as the Nash Equilibrium. John Forbes Nash Jr. in 1950s demonstrated the Finite Game Theory. From that period, the theory is being evolved by the researchers.

It all started evolving after the study of Preemption games in 1980s by Reinganum (in 1981), Fudenberg and Tirole (in 1985), Harris and Vickers (in 1985) and Riordan (in 1992). In the next decade, clock games came into play and the great study improved the computational theories and game design and development a lot [1] [2]. When taken into account the recent works, there's a tremendous increase in the use of automata theory. Qureshi and Mushtaq designed an arcade war game using DFA and NFA in which the player can chose from easy and difficult levels and would be provided with a default weapons and will have to kill the enemies to gather some specific point to reach to another state Q i.e. level, they explained all the levels and the regular expression along with

the input very correctly [3] [4].

In the same year, Qureshi and Zahid explained a roller coaster game using the automata design and this time they came up with some bug fixes and easier concept to design a computer game [5]. The only observable problem was the machines can't be compared to Mealy and Moore machines. To solve this, Abid and Mohsin designed "hungry bird" (a type of game) to explain that game design in Mealy Machine is more understandable than any other kind which can be further extended to more advance automation for future use [6].

The era of static game design is a matter of past, now we have games which have dynamic objects occurring at any time in the middle of game, this paper represents a traditional game in a modernized manner where input will be randomly produced and the obstacles rate too. This type of random strings can be produced with the help of some specific codes in different programming languages and the work can be organized on both computer and android. Android platform has now been evolved as a gaming one for children and youngsters too. This paper gives a solution for the way how the game theory and automaton tools can be used to easily program for a dynamic game.

III. PROPOSED METHOD

Game Description

- Minimum of 2 players and maximum of 4 can choose among the given 3 difficulty levels namely easy, medium and hard from the set Σ_{level} .
- The players are provided with a dice in the game and as per the chance of players when they click on the dice, it will rotate and a random output among the Σ_{dice} set.
- For any player, the game will start when he/she will get a six (6) from the Σ_{dice} set. There is also a benefit of getting 6, the player will get a chance to click the dice again.
- When p1 among the $\Sigma_{players}$ set has clicked then, p2 will get a chance to click the dice and the rules will be same for him too.
- Moving further in the game, they will face obstacles from $\Sigma_{obstacles}$ set. When 'Sn' is the obstacle the player will go some rows/blocks back and when 'St' comes they will gain some rows/blocks.
- After going through everything, the one who reaches 100th block first will be declared as the winner. Σ_{blocks} contains all the blocks where player can move.

Input alphabets

- $\Sigma_{players} = \{p1, p2, p3, p4\}$
- $\Sigma_{level} = \{e, m, h\}$
- $\Sigma_{dice} = \{1, 2, 3, 4, 5, 6\}$
- $\Sigma_{blocks} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, \dots, 95, 96, 97, 98, 99, 100\}$

- $\Sigma_{obstacles} = \{Sn, St\}$

Description of input alphabets

- $\Sigma_{players}$: It defines the set of players (i.e. Maximum 4) who can play the game at a same time. One among the players will start the game and one among the same will win the game.
- Σ_{level} : The player after discussion can choose any difficulty level from the three provided in this set.
- Σ_{dice} : The dice contains six sides containing first six natural number and after clicking on the dice the player will get output among the elements in this set.
- Σ_{blocks} : The board of traditional snakes and ladders game contains counting from 1 to 100 blocks, so does this game contains. The blocks are the spaces where the player can move.
- $\Sigma_{obstacles}$: Snakes and stairs are the obstacles. The former reduces the current block on which the player is and the later increases the chance of the player to win as he/she goes near to the WIN state.

States

TABLE 1: STATES USED IN THE GAME DESIGN

RanInp1	(Initial state) Random input for the first time to initialize the game for any player.
RanInp2	Random input whenever the dice is clicked in the game after start.
Blk	Block which can either increase in number or can decrease.
Rw	Increase and decrease in rows depending upon the obstacles.
WIN	(Final state) which displays the winner's name.

Algorithm of the game

- STEP 1: initialize a variable integer dice=0 and chose a player
- STEP 2: random input in dice from 1 to 6
- STEP 3: if dice is equal to 6, then go to STEP 4, else change player and go to STEP 2
- STEP 4: (again) random input in variable dice from 1 to 6
- STEP 5: if dice is equal to 1 then move one block forward else if dice is equal to 2 then move two blocks forward else if dice is equal to 3 then move three blocks forward else if dice is equal to 4 then move four blocks forward else if dice is equal to 5 then move five blocks forward otherwise dice is equal to 6 then move six blocks forward and go to STEP 4 for an extra chance
- STEP 6: if block reached is Snake, then move specified block backward else if block reached is Stair, then move specified block forward
- STEP 7: change the player again and repeat from STEP 4, STEP 5, STEP 6 and STEP 7 until final block i.e. 100 is not reached

Step 8: exit the game

Now, the process of the game is designed, let's have a look at the flowchart of the game as we know that algorithm and flowchart are the basic building blocks of any complex or easy program. If we are able to design these two then the problem can be solved using codes and can be executed properly. The flowchart can be explained using the algorithm. After the the flowchart we will see the automaton design of the game and will try to understand it more accurately.

Flowchart of the game

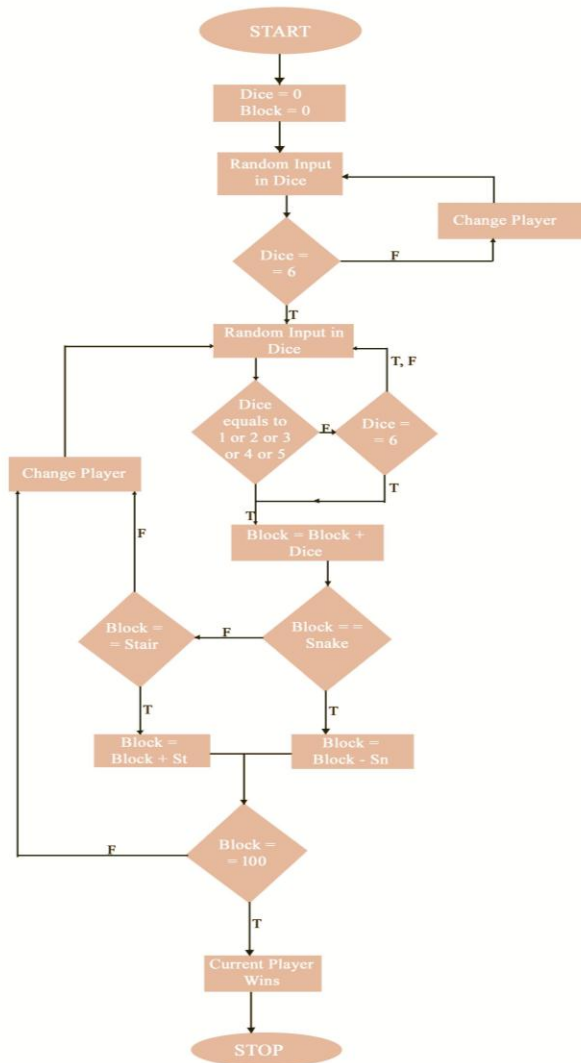


FIG 1: Flowchart of the game design

The difficulties of the game different in the number of rows that will get reduced and increased when the player will land on a snake and a stair respectively. Depending upon this the difficulties are:

1. Easy: When the player will land on a snake he/she will

have to move 20 blocks backward and if there is a stair then they can move 20 blocks forward.

2. Medium: In this, when the player will land on a snake he/she will have to move 30 blocks backward and if there is any stair then to move 20 blocks forward.

3. Hard: This is a mix type of difficulty which will randomly increase the blocks to move when player lands on a stair and will randomly decrease the blocks to get back when the player will land on any snake.

e.x. -> At some moment when the player lands on snake, block to go back is 12 or when on snake block to go forward is 17 and at the other moment the player have to move 20 forward and 5 backward when he lands on stair and snake respectively.

The randomization of the number of blocks to be moved backward and forward will be given through the codes both for android game and computer games.

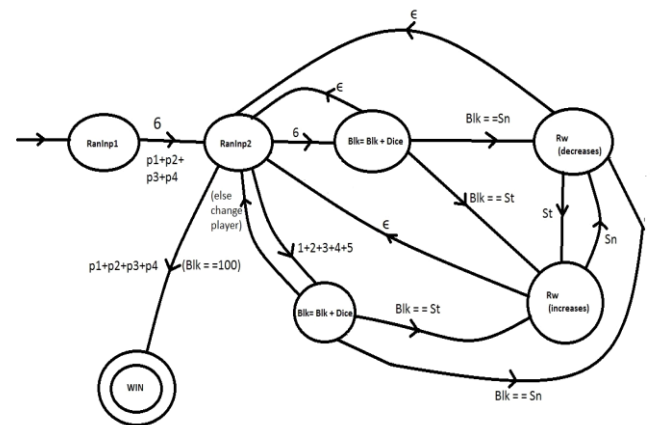


FIG 2: Easy level automaton of the overall game design

The above automaton explains that whenever any player (among p1 or p2 or p3 or p4) will get a 6 from the dice the he will have the chance to start the game and move forward in it. Suppose, he again gets a 6 then he will move 6 blocks forward and will automatically get another chance to throw/click the dice, again getting six will give him a chance again. But if he gets number other than 6 i.e. 1 or 2 or 3 or 4 or 5 then he will move that much block forward and then the other player will get change. Before the other player gets the chance to throw/click the dice the current player will check if he has landed on the stair or the snake, afterwards he will increase or decrease the number of blocks respectively. This will go on until any player reaches the final block i.e. block 100th and hence he will be the winner.

Now let's see what if the player gets only 6 from the dice when he is playing easy one.

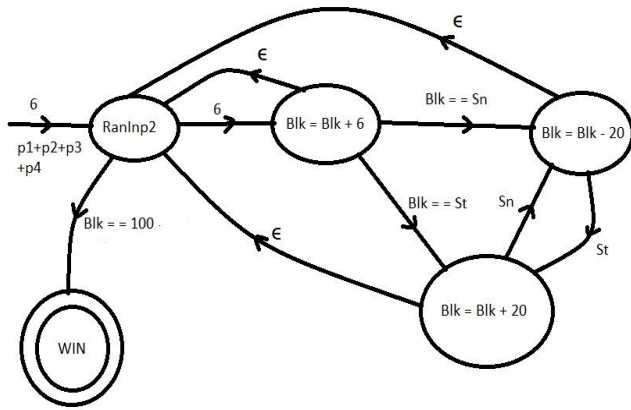


FIG 3: Easy level automaton of the overall game design when any player gets a six on the

When such condition occurs that the player only gets 6 whenever he throws/clicks the dice then only the current player will have the chance to throw/click the dice and if he reaches the 100th block (which is quite difficult as 100 isn't the multiple of 6) then he will Win the game without any other player even starting their game for the first time.

Let's see what happens when the player gets any other number but not a six in an easy level:

TABLE 2: BASIC BOARD DESIGN WHICH WILL EXPLAIN THE BLOACKS AND THE POSITION OF THE BLOCKS AND THE POSITION OF THE SNAKE AND STAIR

100 FINISH	99 SNAKE	98	97	96	95	94	93 SNAKE	92	91
81	82	83	84	85 SNAKE	86	87	88	89	90
80	79 SNAKE	78	77	76	75	74	73	72	71
61	62	63	64 STAIR	65	66	67	68	69	70
60	59	58 STAIR	57	56 SNAKE	55	54	53	52 SNAKE	51
41	42	43	44	45	46	47 SNAKE	48	49	50
40	39 STAIR	38	37	36	35	34	33	32	31
21	22	23 SNAKE	24 STAIR	25	26	27	28 STAIR	29	30
20	19	18	17	16 STAIR	15	14	13	12	11
1 START	2	3	4	5	6	7	8	9 STAIR	10

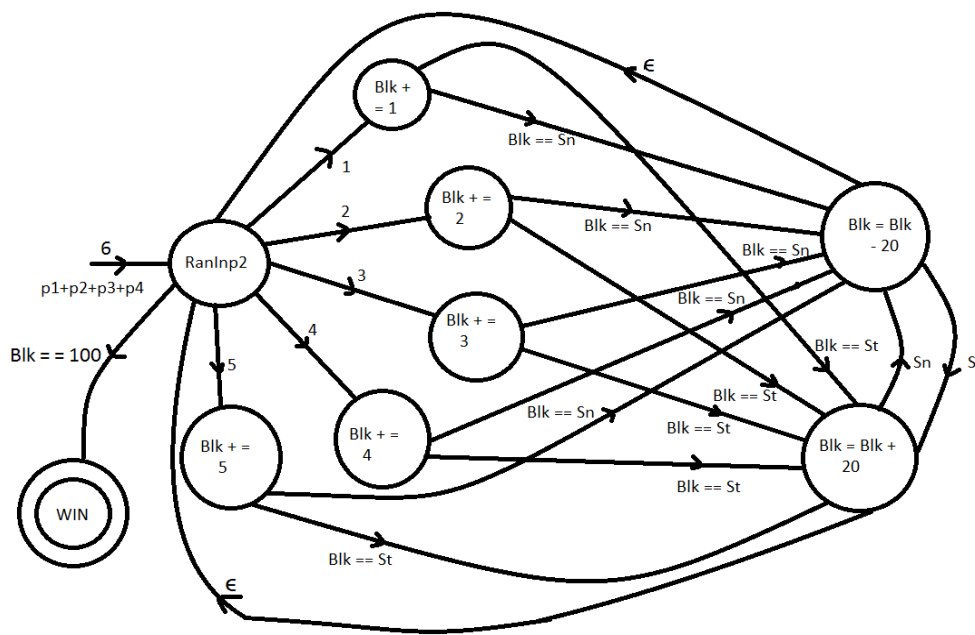


FIG 4: Easy level automaton of the overall game design when any player gets other than a six on the dice

The above automaton shows that the player can reach the Win state by getting 1 or 2 or 3 or 4 or 5 and going through the obstacles too. When there is no obstacles in the path then after the chance player gets changed which hasn't been shown due to easiness. Here, as its easy, the blocks decreasing and increasing is only 20 but, in case of medium difficulty the case is different and it will be quite hard for the player to reach the Win state whereas, in hard level the player might win easily if he has luck otherwise he may never win (possibly).

The same design can be made both for the medium and the hard level of the game.

IV. CONCLUSION

This paper briefly explains the use of game theory and automaton tools along with flowchart to easily get the idea behind designing any game for computer as well as smartphones. These techniques can be used to solve the real life problems all using the wide but easy field of automatas. Further work is going on in the development of the Snakes and Stairs Game in the same manner as it has been explained in this paper and this is going on for the android platform. The advantage of using these techniques is that it provides an easy reference inside the problem and a quick method in solving that. Moreover, the incorrect input can be accepted, as Finite Automata has been used, which validates the input string and runs only for the strings which can give any required output.

V. FUTURE SCOPE

To solve the real life problem as well as in designing any software for military, educational, medical purposes. This paper shows that the techniques can be used to randomize the input and get any input just like an artificial intelligence system which works on itself and randomizes the output when a specific input is sensed.

References

- [1] J. C. Harsanyi, R. Selten, J. W. Weibull, E. V. Damme, J. F. Nash, P. Hammerstein, H. W. Kuhn, "The Work of John Nash in Game Theory", in journal of economic theory Economic Sciences, Vol.12, Issue.1, pp.161-190, 1994.
- [2] M. K. Brunnermeier, J. Morgan, "Clock Games: Theory and Experiments", Games and Economic Behavior., Vol.68, Issue.2, pp.532-550, 2010.
- [3] NS Qureshi, H Mushtaq, MS Aslam, M Ahsan, "Computing Game Design with Automata Theory", International Journal of Multidisciplinary Sciences and Engineering, Vol.3, issue.5, pp.13021, 2012.
- [4] Abid Jamil, "An Infinite Runner Game Design using Automata Theory", International Journal of Computer Science and Software Engineering, Vol.5, Issue.7, pp.119-125, 2016.

- [5] NS Qureshi, Z Abbas, M Sohaib, M Arshad, "A Roller Coaster Game Design using Automata Theory", International Journal Of Multidisciplinary Sciences And Engineering, Vol.3, Issue.5, pp.40-45, 2012.
- [6] A. Jamil, E. A. Ullah, M. Rehman, "An Infinite Runner Design using Automata Theory", IJCSSE, Vol..5, Issue.7, pp.119-125, 2016.

AUTHORS PROFILE

Nishchal Raj pursuing B.E. degree in C.S.E., currently in 4th Sem, at LNCT College from RGPV University. Other than the research, his interests include programming on competitive sites and for android apps along with editing the contents and photos for making them more attractive.



Ratnesh Kumar Dubey received B.E. Computer Science & Engineering from RGPV University Bhopal in 2007. M. tech. Computer Science & Engineering from RGPV University Bhopal in 2011, presently he is working as Assistant Professor in the department of CSE, LNCT Bhopal, India. Research Interest in Software Engineering, Software Testing, Wireless Network, Image Processing. 15 International journals Publication in Reputed Journal and 05 National Journal, 03 IEEE International conferences and 06 National Conference paper Published

