Automata-Based Approach to Ludo Game Design and Development using JFLAP

A. Neeraja
dept. of Computer Science
Amrita University
Bangalore
bl.en.u4cse22004@bl.students.amrita.edu

Ch. Mounika Begum
dept. of Computer Science
Amrita University
Bangalore
bl.en.u4cse22014@bl.students.amrita.edu

K. Vigneswara Reddy
dept. of Computer Science
Amrita University
Bangalore
bl.en.u4cse22030@bl.students.amrita.edu

R. Henry Koushal
dept. of Computer Science
Amrita University
Bangalore
bl.en.u4cse22050@bl.students.amrita.edu

***Abstract*— In the scope of compiler construction, programming languages, image compression, bioinformatics, and a morphological analysis to some extent, computation and automata theory is fundamental. This study focuses on computing and automata with design charts for a Ludo game. The steps of Ludo development process include the use of DFA (Deterministic Finite State Automata) and NDFA (Non-Deterministic Finite State Automata) across the various tiers. By this thorough investigation, we clarify how these primary conceptions are channeling to build unique and thrilling games.**

Keywords-NDFA and DFA, Automata theory, Ludo, Game design, JFLAP.

I. Introduction

Alan Turing who is also remembered as the brilliant scientist and the founder of computational thinking, defined the automata theory and the idea of 'Turing machines', which had solved the majority of the problems that Hilbert and other mathematicians had been dealing with [1]. Such fundamental instruments drastically changed the field of computing and are very much necessary in the video game industry as well. The impact of computation and automata, mainly in computer video games, but other media forms as well, has been multifaceted.

Despite the fact that the core of automata theory is within the reach, the extent of its application, paired with game theory, for creating computer video games is huge where most of the potential is left unused. This paper is focused on using the deterministic finite automaton (DFA) and non-deterministic finite automaton (NFA) in our the making of a Ludo game [2], a game that many kids do enjoy. With their childhood memories. Ludo organizes up to four players, who just roll a dice with the numbers one through six, and try to run their four pawns to the final. The game's main idea is the selection of moves in a strategic style. Moreover, it is possible to win by eliminating the opponent's pawns. This dissertation explains about how basic automata concepts can be employed for the improvement of the structure and physics of the most cherished games.

II. Literature Survey

Deterministic finite automata are utilized in developing automated evaluation systems. Suggested are efficient procedures for automaton conversion and reduction. A mathematical framework for card swipe machine transactions is presented in the document.

Utilize a Pachisi game to illustrate triadic relational dynamics within supply networks.

Learning involvement encompasses emotional conditions and mental activities. Game-based learning involvement is a unified and ongoing procedure. Academic scrutiny concentrated on conceptualizing game-based learning involvement.

The historical background of the board game Pachisi, which gained global popularity, was initially introduced to England in 1896. The development of games in present-day times necessitates the integration of both game theory and computational theory in their evolution.

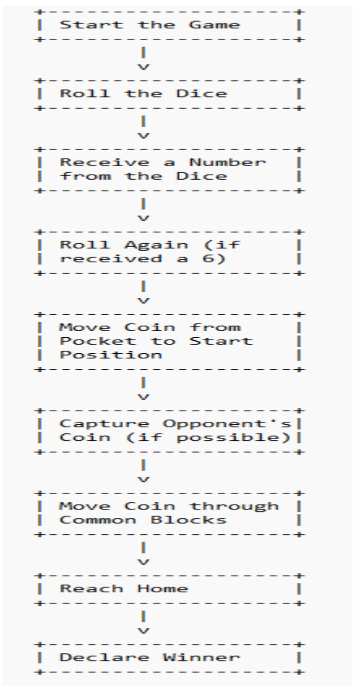
PCG autonomy and player experience modeling are included in the literature study. Game AI classes are assigned readings from the existing literature.

The main goal of the paper is to model an abstract machine that resembles a ladder and a snake.

explains the SNL machine's computational features and uses.

III. Methodology

- ***Workflow***


Start the Game
Roll the Dice
Receive a Number from the Dice
Roll Again (if received a 6)
Move Coin from Pocket to Start Position
Capture Opponent's Coin (if possible)
Move Coin through Common Blocks
Reach Home
Declare Winner

A. Defining the game by automata theory:

Input alphabets:

$\sum_{player}$ ={p1, p2, p3, p4}

$\sum_{dice}$ ={A, B, C, D, E, F}

$\sum_{coinstate}$ ={cp1, cp2, cp3, cp4, o}

$\sum_{coinhome}$ ={h1, h2, h3, h4}

$\sum_{coin}$ ={ fc, sc, tc, ftc}

$\sum_{currentplayer}$ ={op, ap}

B. Description of input alphabets

$\sum_{player}$: This set defines about players who can play the game. Maximum of 4 and a minimum of 2 players can play at a time.1 player among them can win the game at the same time. p1, p2, p3 and p4 are players in the game.

$\sum_{dice}$: A set of dice contains 6 natural numbers when dice are rolled/clicked by the player. The player will receive a randomoutput from the set of dice.1,2,3,4,5and 6 are natural numbers in the set of dice where, we represent them as A, B, C, D, E , F in transition diagram.

$\sum_{coinstate}$: This set tells us about the current state of the coin. It clearly tells us about whether the coin is in the pocket or not.

Representation of elements in $\sum_{coinstate}$:

cp1 =coin is in the pocket of player1

cp2 =coin is in the pocket of player2

cp3 =coin is in the pocket of player3

cp4 =coin is in the pocket of payer4

o =coin is present in any block of Ludo board

$\sum_{coinhome}$: This set defines the coin of a player which reached to home

Representation of elements in $\sum_{coinhome}$

h1 =first coin of player reached home

h2 =second coin of player reached home

h3 =third coin of player reached home

h4 =fourth coin of player reached home

$\sum_{coin}$: This set defines the coins of players.
Representation of elements in $\sum_{coin}$:
fc =first coin of player
sc =second coin of player
tc =third coin of player
ftc =fourth coin of player

$\sum_{currentplayer}$: This set defines the status of the player whether he is an active player or opponent player.

Representation of elements in $\sum_{currentplayer}$:

ap =active player

op =opponent player

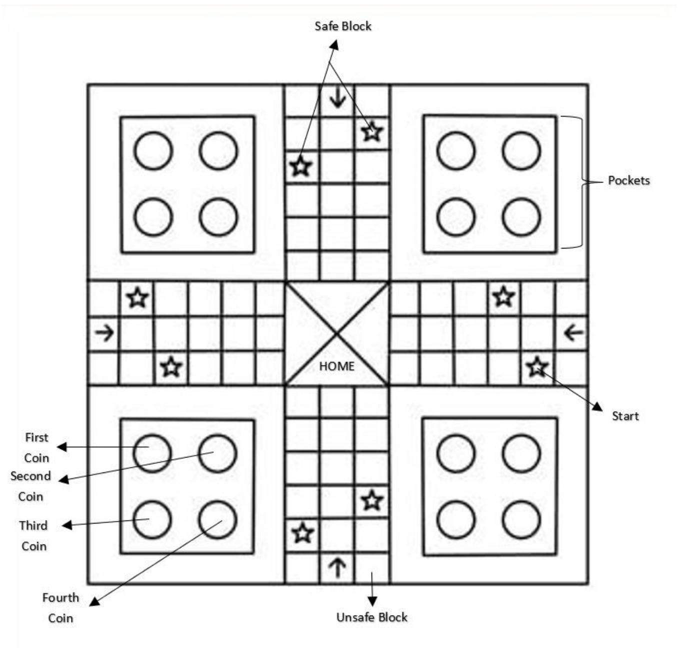


Fig. 2. Overall design of Ludo board

▪ ***Algorithm for Game to Develop Automata Theory***

1. Initialize the Game:
    - Set the dice value to 0.
    - Distribute all the coins among the pockets of the players.
2. Player Move:
    - Roll the dice or click.
    - If the dice value is 6:
        - The active player can pick any coin from their pocket and place it on the start position.
        - Go to the next step.
    - Else:
        - Switch the active player.
        - Go back to step 2.
3. Coin Move:
    - Roll the dice or click again.
    - Depending on the dice value:
        - If the value is 1, 2, 3, 4, 5, or 6
        - Move any one of the active player's coins forward by the number of blocks based on the value.
        - If the value is 6, the player gets an extra turn back to step 3.
        - If the coin reaches an unsafe block (already occupied by an opponent's coin), proceed to step 6.
        - Go back to step 2.

4. Unsafe Block case:
   - If the active player’s coin reaches an unsafe block (occupied by an opponent’s coin):
   - Send the opponent’s coin back to their pocket.
   - Continue with the next player’s turn (go

     back to step 2)
5. Repeat Turns:
   - Continue alternating turns between players (steps 2-4) until all coins of both players reach the home position.
6. Declare the Winner:
   - If all coins of a player reach the home position:
   - Declare that player as the winner.
   - End the game.
7. Quit the Game:
   - Exit the game.

A. States used in game design:

BF→ The current block of the first coin. the coin can move forward with the output of dice.

BS→ The current block of the second coin. the coin can move forward with the output of dice.

BT→ The current block of the third coin. The coin can move forward with the output of dice.

BFO→ The current block of the fourth coin. The coin can move forward with the output of dice.

Z→ Coin moves forward to a block when a player receives 1 or 2 or 3 or 4 or 5 on dice

Coin Reached Home(CRH)→ Coins of players which reached home

Random Output1(RI 1)→ The random output from dice when rolled/clicked by the player to start the game

Random Output2(RI 2)→ The random output from dice when rolled/clicked by the player after the start of the game to move coins.

WIN→ Which displays the winner player (Final State)

- ***Design and Explanation of Automatons***

From figure 2, automaton explains whenever player among (p1+p2+p3+p4) receives 6 can start the game by taking out one coin from pocket to start positioning it this has been represented in the automata as p1F,p2F,p3F,p4F where F represent 6 on dice whenever a player with input 6 occurs then they should start the game. As player received6, he will have an extra chance to roll the dice. If he receives6 again on dice he can get a coin out from pocket to start position or he can move any coin forward by six blocks and receives an extra chance to roll dice. In above automaton, if first coin of active player is in pocket and receives six on his dice as Fcp1fc then he can receive first coin from pocket to start position [reaches state ‘BF=start’ in automaton] in the same way it continues till receiving fourth coin.

Similarly, on receiving six on dice, player can receive one of remaining coin in pocket to start position. Player can move first coin (Fofc) forward by 6 blocks from current block [reaches state BF+=6]. If the player receives 1 or 2 or 3or 4 or 5 or 6 reaches ‘z’ in automaton where this has been represented as A, B, C, D, E, F. Figure 4 to 8 shows possibleNFA’s when the player receives the number other than 6 on his/her dice for moving of first coin or second coin or third coin or fourth coin.

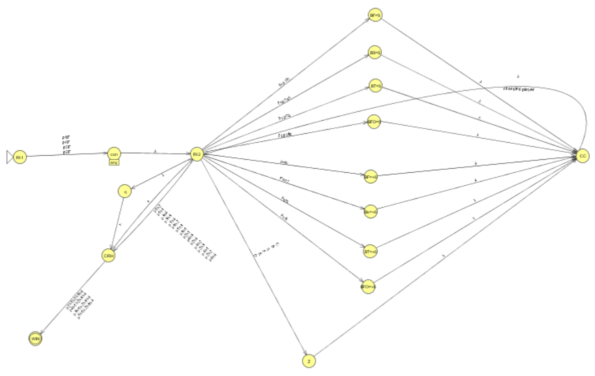


Fig. 2. Macro-Level Automaton of Overall Game Design

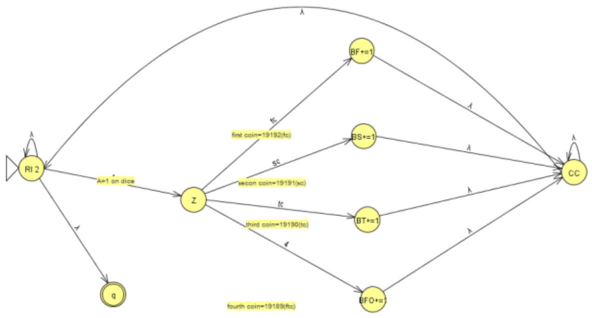


Fig. 3. Player received 1 on dice

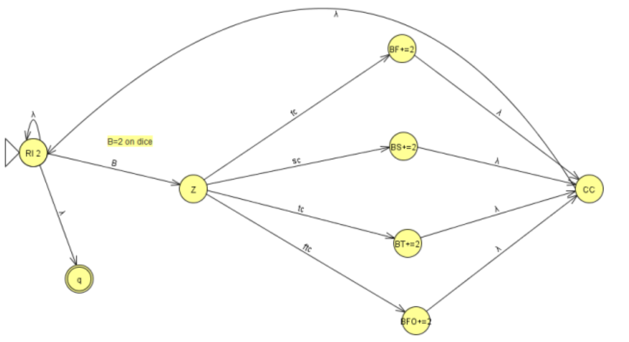


Fig. 4. Player received 2 on dice

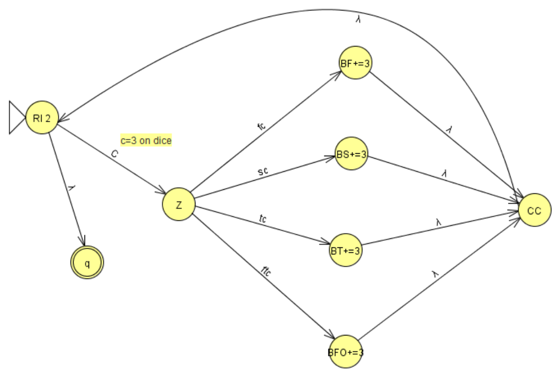


Fig . 5. Player received 3on dice

From figure 3, automata represent when any player receives 1 on his dice then player can move any one of the coins by 1 block forward. Similarly, we can easily explain figures 4, 5, 6, 7, 8. As coin move towards another block, it checks whether the block is occupied by the coin of opponent player. If the block is occupied by another player then coin of opponent player is sent back to the pocket. Figures 9 to 12 shows possible NFA's when the coin of active player reaches a block containing coin which might belong to different opponent players.

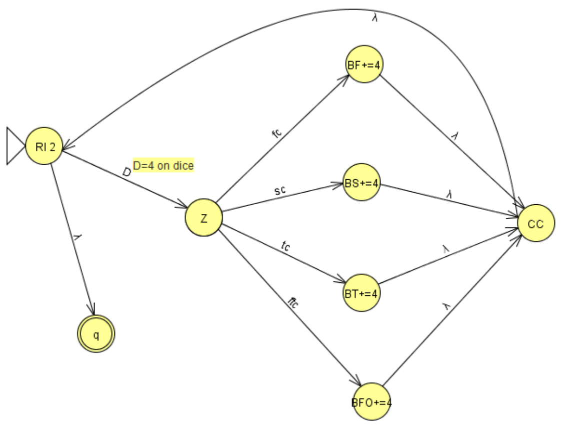


Fig .6. Player received 4 on dice

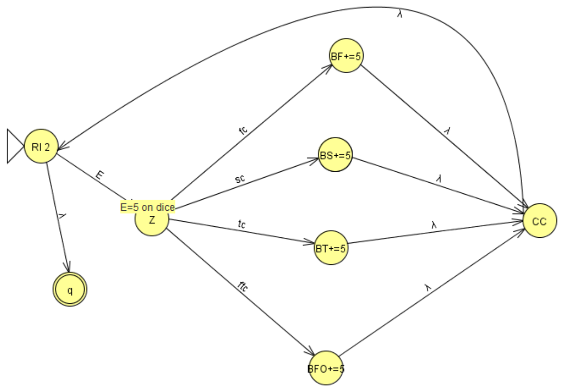


Fig. 7. Player received 5 on dice

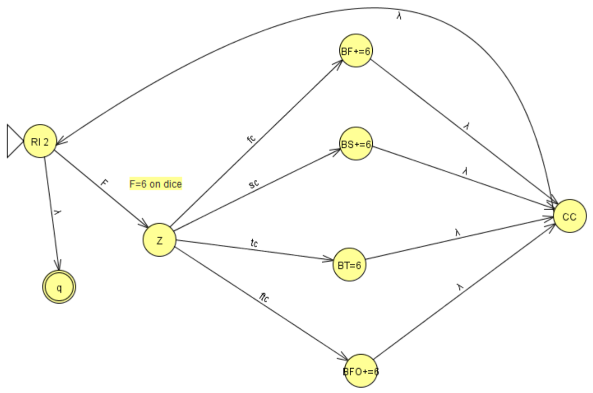


Fig. 8. Player received 6 on dice

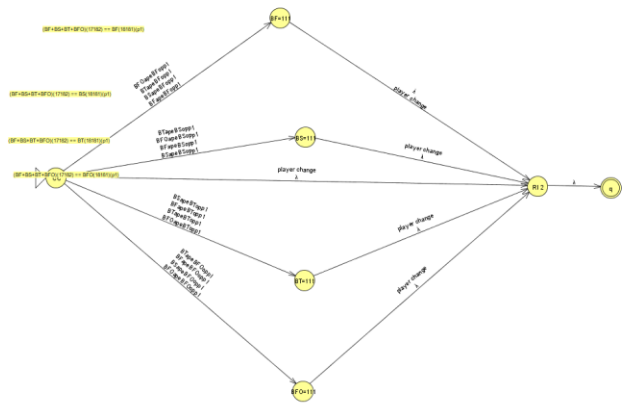


Fig. 9. The active player coin reaches a block having player 1's coin

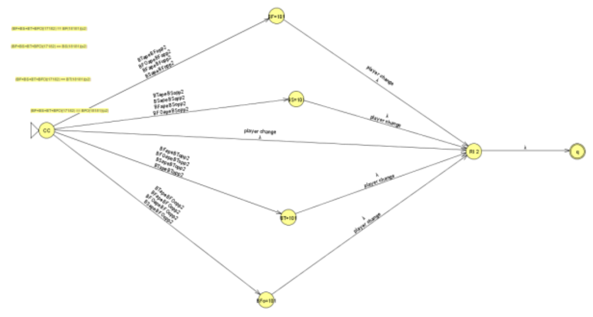


Fig. 10. The active player coin reaches a block having player 2's coin

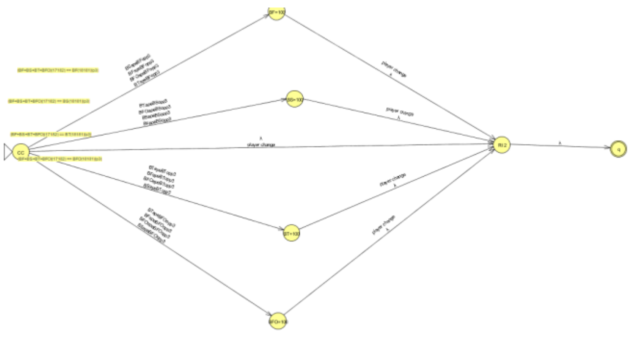


Fig. 11. The active player coin reaches a block having player 3's coin

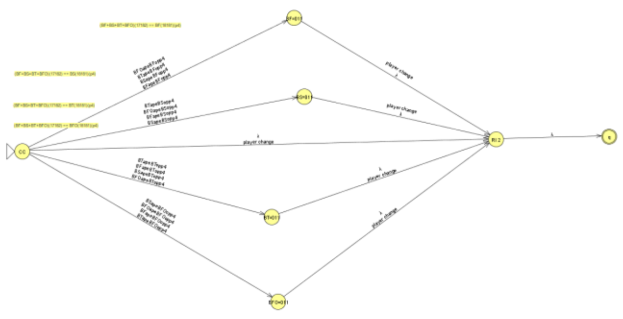


Fig. 12. The active player coin reaches a block having player 4's coin

In figure 9, when automaton reaches capturing coins it checks whether any coin moved by the active player to a block containing first coin or second coin or third coin or fourth coin of player 1. The block to which coin of the active player is moved contains any coin of player1 then the respective coin is sent to player 1's pocket. Similarly figure 10, figure 11, figure12 can be easily explained. After capturing coins, if the last random output of dice is 6 then goto random input2 for extra chance else change the active player. This will go on until all coins of any player reach home. The player whose all coins reaches home will be declared as the winner.

- ***Tracing***

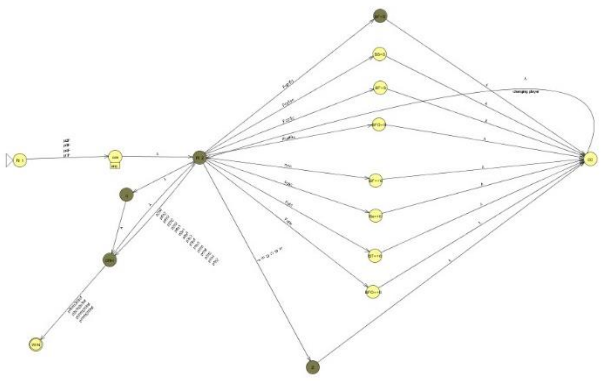


Fig. 13. The tracing and working of overall design Without reaching to final state

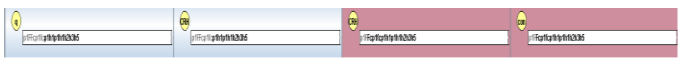


Fig. 14. String Rejectance

In Figure 13, we trace through the top-level design of the Ludo game. The game them moves to next states but it is not reaching final stage because the input string is not provided in proper format.

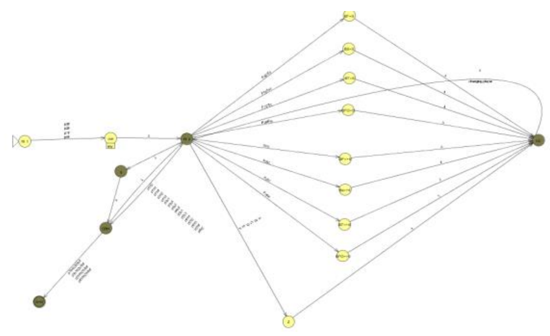


Fig. 15. The tracing and working of overall design when reaching to final state

Fig. 16. String Acceptance

In Figure 15, we trace through the top-level design of the Ludo game. The user rolls the dice, resulting in the random output state RI 2. The game then moves on through the game until such time as coins collide and are captured at cc. Then, using a lambda transition, the game returns to the RI 2 state, proceeding to an intermediate state labeled 'q.' Then the player's coins move home according to the game's rules. Finally, when all the player's coins are home, the game transitions to a final state, to denote that the player has won the game. As we trace through this, we darken in all the critical states and transitions so that we can see the top-level game progression.

B. *Transition tables:*

Player received 1 on dice								
State	RI2	Z	BF+=1	BS+=1	BT+=1	BFO+=1	CC	q
RI2	λ	A	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=1	–	–	–	–	–	–	λ	–
BS+=1	–	–	–	–	–	–	λ	–
BT+=1	–	–	–	–	–	–	λ	–
BFO+=1	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Player received 2 on dice								
State	RI2	Z	BF+=2	BS+=2	BT+=2	BFO+=2	CC	q
RI2	λ	B	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=2	–	–	–	–	–	–	λ	–
BS+=2	–	–	–	–	–	–	λ	–
BT+=2	–	–	–	–	–	–	λ	–
BFO+=2	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Player received 3 on dice								
State	RI2	Z	BF+=3	BS+=3	BT+=3	BFO+=3	CC	q
RI2	λ	C	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=3	–	–	–	–	–	–	λ	–
BS+=3	–	–	–	–	–	–	λ	–
BT+=3	–	–	–	–	–	–	λ	–
BFO+=3	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Player received 4 on dice								
State	RI2	Z	BF+=4	BS+=4	BT+=4	BFO+=4	CC	q
RI2	λ	D	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=4	–	–	–	–	–	–	λ	–
BS+=4	–	–	–	–	–	–	λ	–
BT+=4	–	–	–	–	–	–	λ	–
BFO+=4	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Player received 5 on dice

State	RI2	Z	BF+=5	BS+=5	BT+=5	BFO+=5	CC	q
RI2	λ	E	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=5	–	–	–	–	–	–	λ	–
BS+=5	–	–	–	–	–	–	λ	–
BT+=5	–	–	–	–	–	–	λ	–
BFO+=5	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Player received 6 on dice

State	RI2	Z	BF+=6	BS+=6	BT+=6	BFO+=6	CC	q
RI2	λ	F	–	–	–	–	–	λ
Z	–	–	fc	sc	tc	ftc	–	–
BF+=6	–	–	–	–	–	–	λ	–
BS+=6	–	–	–	–	–	–	λ	–
BT+=6	–	–	–	–	–	–	λ	–
BFO+=6	–	–	–	–	–	–	λ	–
CC	λ	–	–	–	–	–	λ	–
q	–	–	–	–	–	–	–	–

Active player coin reaches block having player 1's coin

State	CC	BF=111	BS=111	BT=111	BFO=111	RI2	q
CC	–	BFapeBFopp1+ BSapeBFopp1+ BTapeBFopp1+ BFOapeBFopp1	BFapeBSopp1+ BSapeBSopp1+ BTapeBSopp1+ BFOapeBSopp1	BFapeBTopp1+ BSapeBTopp1+ BTapeBTopp1+ BFOapeBTopp1	BFapeBFOopp1+ BSapeBFOopp1+ BTapeBFOopp1+ BFOapeBFOopp1	λ+Player change	–
BF=111	–	–	–	–	–	λ+Player change	–
BS=111	–	–	–	–	–	λ+Player change	–
BT=111	–	–	–	–	–	λ+Player change	–
BFO=111	–	–	–	–	–	λ+Player change	–
RI2	–	–	–	–	–	–	λ
q	–	–	–	–	–	–	–

Active player coin reaches block having player 2's coin

State	CC	BF=101	BS=101	BT=101	BFO=101	RI2	q
CC	–	BFapeBFopp2+ BSapeBFopp2+ BTapeBFopp2+ BFOapeBFopp2	BFapeBSopp2+ BSapeBSopp2+ BTapeBSopp2+ BFOapeBSopp2	BFapeBTopp2+ BSapeBTopp2+ BTapeBTopp2+ BFOapeBTopp2	BFapeBFOopp2+ BSapeBFOopp2+ BTapeBFOopp2+ BFOapeBFOopp2	λ+Player change	–
BF=101	–	–	–	–	–	λ+Player change	–
BS=101	–	–	–	–	–	λ+Player change	–
BT=101	–	–	–	–	–	λ+Player change	–
BFO=101	–	–	–	–	–	λ+Player change	–
RI2	–	–	–	–	–	–	λ
q	–	–	–	–	–	–	–

Active player coin reaches block having player 3's coin

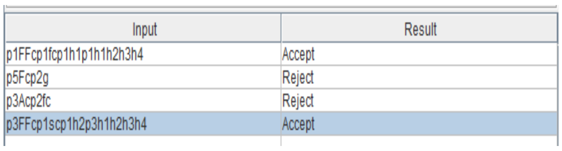
State	CC	BF=100	BS=100	BT=100	BFO=100	RI2	q
CC	–	BFapeBFopp3+ BSapeBFopp3+ BTapeBFopp3+ BFOapeBFopp3	BFapeBSopp3+ BSapeBSopp3+ BTapeBSopp3+ BFOapeBSopp3	BFapeBTopp3+ BSapeBTopp3+ BTapeBTopp3+ BFOapeBTopp3	BFapeBFOopp3+ BSapeBFOopp3+ BTapeBFOopp3+ BFOapeBFOopp3	λ+Player change	–
BF=100	–	–	–	–	–	λ+Player change	–
BS=100	–	–	–	–	–	λ+Player change	–
BT=100	–	–	–	–	–	λ+Player change	–
BFO=100	–	–	–	–	–	λ+Player change	–
RI2	–	–	–	–	–	–	λ
q	–	–	–	–	–	–	–

Active player coin reaches block having player 4's coin

State	CC	BF=011	BS=011	BT=011	BFO=011	RI2	q
CC	–	BFapeBFopp4+ BSapeBFopp4+ BTapeBFopp4+ BFOapeBFopp4	BFapeBSopp4+ BSapeBSopp4+ BTapeBSopp4+ BFOapeBSopp4	BFapeBTopp4+ BSapeBTopp4+ BTapeBTopp4+ BFOapeBTopp4	BFapeBFOopp4+ BSapeBFOopp4+ BTapeBFOopp4+ BFOapeBFOopp4	λ+Player change	–
BF=011	–	–	–	–	–	λ+Player change	–
BS=011	–	–	–	–	–	λ+Player change	–
BT=011	–	–	–	–	–	λ+Player change	–
BFO=011	–	–	–	–	–	λ+Player change	–
RI2	–	–	–	–	–	–	λ
q	–	–	–	–	–	–	–

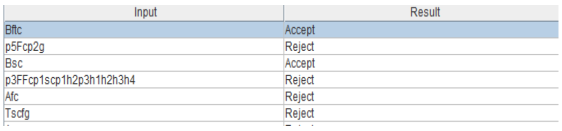
IV. RESULTS

In this scenario, there are 11 NFAs designed for different tasks. Six of them are related to the numbers on a dice, where acceptance depends on the user inputting numbers between 1 to 6 and choosing coins between 1 to 4. If the input matches these criteria, the specific string will be accepted; otherwise, it will be rejected. Additionally, four NFAs check for collisions between active player and opponent player coins. Acceptance or rejection is determined based on this collision logic. Each NFA has been assigned specific tasks and variables to ensure that only certain strings are accepted, while others are rejected**.**



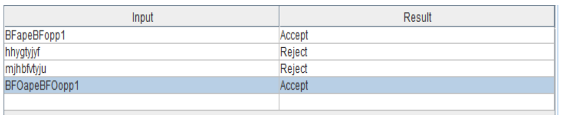
Input	Result
p1FFcp1fcp1h1p1h1h2h3h4	Accept
p5Fcp2g	Reject
p3Acp2fc	Reject
p3FFcp1scp1h2p3h1h2h3h4	Accept

Fig. 13. Overall Game Design



Input	Result
Bftc	Accept
p5Fcp2g	Reject
Bsc	Accept
p3FFcp1scp1h2p3h1h2h3h4	Reject
Afc	Reject
Tscfg	Reject

Fig.14. When no on dice is 1,2,3,4,5,6



Input	Result
BFapeBFopp1	Accept
hhygtyjyf	Reject
mjhbfvtyju	Reject
BFOapeBFOopp1	Accept

Fig.15. whenever the active player coin reaches Players 1,2,3,4 coin

Macro design for overall ludo game

State	RI1	con(constraint)	RI2	BF=S	BS=S	BT=S	BFO=S	BF=+6	BS=+6	BT=+6	BFO=+6	CC	q	z	CRH	WIN
RI1	–	p1F+p2F+p3F+p4F	–	–	–	–	–	–	–	–	–	–	–	–	–	–
con(constraint)	–	–	λ	–	–	–	–	–	–	–	–	–	–	–	–	–
RI2	–	–	–	Fcp1fc	Fcp1sc	Fcp1tc	Fcp1ftc	Fofc	Fosc	Fotc	Foftc	–	λ	A+B+C+D+E+F	p1h1+p1h2+p1h3+p1h4+ p2h1+p2h2+p2h3+p2h4+ p3h1+p3h2+p3h3+p3h4+ p4h1+p4h2+p4h3+p4h4	–
BF=S	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BS=S	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BT=S	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BFO=S	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BF=+6	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BS=+6	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BT=+6	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
BFO=+6	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
CC	–	–	λ+ change player	–	–	–	–	–	–	–	–	–	–	–	–	–
q	–	–	–	–	–	–	–	–	–	–	–	–	–	–	λ	–
z	–	–	–	–	–	–	–	–	–	–	–	λ	–	–	–	–
CRH	–	–	λ	–	–	–	–	–	–	–	–	–	–	–	–	p1h1h2h3h4+ p2h1h2h3h4+ p3h1h2h3h4+ p4h1h2h3h4
WIN	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

V. Conclusion

The proposed work concisely deals the design of the Ludo game using automata theory. The automata theory has been used for designing the designing the multi-player games in more efficiently for PC or smartphones. NDFA is the most powerful concept has used for design of proposed LUDO game which resulted high performance gaming with limited resources. The designed LUDO game using automata resulted bug free, high accuracy, and efficiency for multiplayer.

In our implementation, we tackled numerous challenges while developing the game. We referenced existing papers to guide our efforts and modified NFA (Nondeterministic Finite Automaton) structures to suit our needs. Specifically, we constructed a total of 11 NFAs for various functionalities. Six of these NFAs were dedicated to determining the outcomes of rolling a dice, while four were designed to check for collisions between opponent players' coins in relation to their respective houses. Additionally, we created one NFA to illustrate the overall design and functionality of the game. Through these efforts, we aimed to ensure a comprehensive and effective implementation of the game mechanics.

In conclusion, we have implemented the macro-level overall design of a Ludo game using JFLAP. JFLAP simplifies the implementation of finite automata with its user-friendly interface and visual representation, enabling users to simulate, analyze, and experiment with various automata types. Additionally, its support for exporting/importing models and educational resources enhances its utility as a tool for both learning and practical application in automata theory.

VI. Future Scope

In the future, we aim to tackle real-world problems and develop games using automata theory as a foundation. Additionally, we plan to explore the implementation of these problems using more complex computational models such as pushdown automata and Turing machines, broadening our understanding and applicability of theoretical concepts in practical scenarios.

VII. Refernences

[1] Z Wu, TY Choi Decision Sciences Journal of Innovative Education, 2013.

[2] F Ke, K Xie, Y Xie British Journal of Educational Technology, 2016

[3] KF Ali, V Kalyan, KA Kumar 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), 2019

[4] R Ahmed2021

[5] Arslan, Farrukh, et al. "Enhancing Card Swipe Machines using Mathematical Model with JFLAP Formal Methods and Automation: A Mathematical Model with JFLAP." VAWKUM Transactions on Computer Sciences 11.1 (2023): 108-122.

[6] F. Alvi and M. Ahmed, “Complexity analysis and playing strategies for Ludo and its variant race games,” in 2011 IEEE Conference on Computational Intelligence and Games, CIG 2011, 2011.

[7] K. N. S. Heeroo, O. Gukhool, and D. Hoorpah, “A Ludo Cellular Automata model for microscopic traffic flow,” Journal of Computational Science, 2016.

[8] A. K. Jagannatham and V. Kumar, “Introduction to game theory,” in Decision Sciences: Theory and Practice, 2016.

[9] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” IEEE Transactions on Parallel and Distributed Systems, 2015.

[10] M. K. Brunnermeier and J. Morgan, “Clock games: Theory and experiments,” Games and Economic Behavior, 2010.

[11] P. Blackburn, M. de Rijke, and Y. Venema, “Computability and Complexity,” in Modal Logic, 2014.

[12] 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, 2012, pp.13021.

[13] bid 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.

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

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

[16] J.M Markus, K.Brunnermeier “Clock Games:Theory and Experiments”, 2010, pp.348-363.

[17] K. F. Ali, V. Kalyan and K. A. Kumar, "Design and Implementation of Ludo Game Using Automata Theory," 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2019, pp. 1-6, doi: 10.1109/i-PACT44901.2019.8959998.