

A Roller Coaster Game Design using Automata Theory

Noman Sohaib Qureshi¹, Zahid Abbas², Muhammad Sohaib², Muhammad Arshad², Rizwan Ali Sabir², Asma Maqsood²

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

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

Abstract—Automata Theory is a useful tool in multidisciplinary computing and scientific research. Game design is also an area where Automata theory may be applied successfully. Video games are an automaton itself. In computing, every software application is automata behind the scene working on the principle algorithms that automates its execution. This paper presents the design of a roller coaster game.

Keywords– Automata Theory, Game Design and Roller Coaster Game

I. INTRODUCTION

A game designed using ‘automata and computer theory’ tools is easier to code and develop rather than using any other tool. Any software in computer itself works on the principle of automata or automation. More and more disciplines where computing is used are using in their research the computation and automata tools that help in a great manner solving complex problems with ease. Game design is no exception. In this paper a roller coaster game is designed using various automata tools. Non-deterministic finite state automata are used for the preliminary design which will eventually be converted to deterministic finite state automata for development and coding. Experimental run of the

automata machine is carried out and shown for the roller coaster game in this paper.

II. EXISTING GAME THEORY REVIEW

Game theory is a field of applied mathematics that describes and analyzes interactive decision situations. It provides outcome of complex interactions among rational entities, where rationality demands strict adherence to a strategy based on perceived or measured results. The main areas of application of game theory are economics, political science, biology and sociology. From the early 1990s, engineering and computer science have been also included in the game theory [1].

It addresses the interaction among individual rational decision makers. Such models are called ‘games’ and the rational decision makers are referred to as players. Game theory is the branch of decision theory concerned with interdependent decisions. The problems of interest involve multiple participants, each of whom has individual objectives related to a common system or shared resources [2].

There are many sub disciplines in game theory. The boundaries among them are not firm but fuzzy and there is a considerable relevancy involving substantive areas such as law or economics. Conversational as well as formal game theory still abounds and this is not merely indicative of preformal investigation prior to formal modeling; but even more importantly it shows that in everyday business and politics the language of game theory is Influencing practitioners and the public in general about the concepts of strategic analysis. As von Neumann intimated any good applied mathematics has its assumption or axioms rooted in observation. The relationship between context and the concept of solution is critical [3].

Game theory has been used as a tool to study different aspects of computer and telecommunication networks, primarily as applied to problems in traditional wired networks. Since the game theoretic models, developed for adhoc networks, focus on distributed system’s results and conclusions generalize as well as the number of players (nodes) is increased [4]. Emerging research in game theory applied to ad hoc networks shows much promise to help understand the complex interactions between nodes in this highly dynamic and distributed environment. Behavioral

Noman Sohaib Qureshi is with the Department of Computer Science & Engineering, University of Engineering and Technology, Lahore (Gujranwala Campus-RCET), (phone: +923009682419; e-mail: nomee_46@hotmail.com).

Zahid Abbas is with the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), (phone: +923438641362; e-mail: m.zahid100@gmail.com).

Muhammad Sohaib is with the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), (phone: +923447077002; e-mail: sohaib_136@yahoo.com).

Muhammad Arshad Jahangir is with the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), (phone: +923457524574; e-mail: arshadjahangir@ymail.com).

Rizwan Ali Sabir is with the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), (phone: +923074301898; e-mail: rizwanalisabir@gmail.com).

Asma Maqsood is with the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore (Gujranwala Campus-RCET), (phone: +923007595016; e-mail: asma.maqsood2010@yahoo.com).

models of games are also meant to be general, in the sense that the same models can be applied to many games with minimal customization. A simple model of thinking steps attempts to predict choices in one-shot games and provide initial conditions for learning model. Game theory can help explain naturally-occurring phenomena [5].

In gaming, players' actions are referred to as moves. The role of analysis is to identify the sequence of moves that you should use. A sequence of moves is called a strategy, so an optimal strategy is a sequence of moves that results in the best outcome [6].

Thomas Schelling [7] brings game theory to life, and does so using two varied approaches. First, instead of the abstract and austere beauty of mathematical theorem's that we find in most of game theory going back to von Neumann, Morgenstern and Nash, he gives us lively and memorable examples to explain and illustrate his idea. He elucidated many of the most basic concepts and strategies of game theory—credibility, commitment, threats and promises, brinkmanship, focal points and the tipping of equilibrium, to mention just a few. These ideas became the foundations for numerous applications in economics. He has given us game theory to practice, and game theory to remember. Originally, von Neumann and Morgenstern developed game theory as a mathematical method especially adapted to economics and social science in general. In the introduction of their book, they emphasized their view that methods taken over from the natural sciences are inadequate for their purpose. Aumann has forcefully argued that a game theoretic solution concept should be judged primarily by the insights that it yields in applications, by "its success in establishing relationships and providing insights into the workings of the social processes to which it is applied" [7].

III. ROLLER COASTER GAME DESIGN

PROBLEM STATEMENT:

Roller coaster with various states is designed as follows:

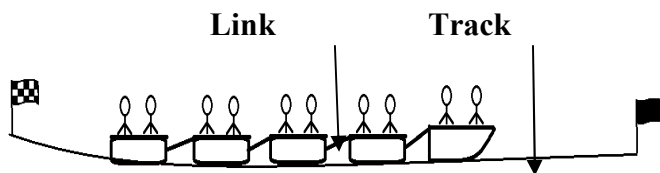


Fig. 1: Roller Coaster Model

- The initial stage is with 'black and white' flag
- The final stage is with 'black flag'

The target is to take roller coaster to the final stage by making a track which will satisfy the must conditions in the design. Also when roller coaster reaches the final stage, the links should be unbroken.

A. Initial State:

Black & White Flag denoted by "B\W flag"

B. Final State:

Black Flag denoted by "B.F"

C. Input Alphabets:

The input alphabet has symbols and functionality in pattern containing name, symbols and functions.

$$\Sigma = \{\Lambda, B, S, F, c, but, sdt, rt, ht, ec, sdp, z, ps, sut\}$$

In the sigma, 'c4' is translated into 'cccc' that is four times a coin collected and similarly input alphabet: c5, c6, c7 is translated into respective number of c occurrences and that is five, six and seven times a coin is collected.

1) Roller coaster

The roller coaster is represented by *rc* which is the basic default input alphabet needed to start with. It is obvious that before going from one state to other, the special input alphabet "rc" is required. At that stage, to design finite state automaton (FSA) simply the assumption is made that this input string from Σ will be there at the required directed edge of the FSA.

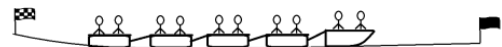


Fig. 2: Roller Coaster with $\Sigma = 'rc'$

2) Broken Track (B)

This input shows that path is broken and roller-coaster will be unable reach final track. Moreover, it is at any state and read the input alphabet "B" the speed of roller coaster will be 0km/hr and this input alphabet is used when a player is not willing to win the game or is forcedly used by the player when he has no experience that how to make the path, whenever this input string is read from Σ the player will be in lose state.



Fig. 3: Roller Coaster with Broken Track

3) Sharp Edge (S)

This sigma element will break the roller coaster.

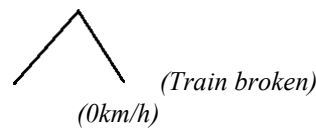


Fig. 4: Roller Coaster with Broken Track

Also when a player is at any state and read the input alphabet "S" the effect will be the same as was with the input alphabet "B" that is the speed of roller coaster will be 0km/hr and this input alphabet is used when certain player is not willing to win the game or is forcedly used by the player when he has no experience that how to make the path towards winning states. By reading this alphabet it will never be able to reach some final state.

Moreover when a certain player is at any state and reads the input alphabet “z” the speed of roller coaster will increase and this input alphabet is used when the player have to collect the coins at zigzag place or position or it is used where roller coaster needs acceleration to reach the final state by collecting the coin at zigzag place.



(on handmade track, 15km/h).

Fig. 14: Energy Capsule for Roller Coaster at Σec



Fig. 11: Coin Collection on Zigzag Track at 10 km/h

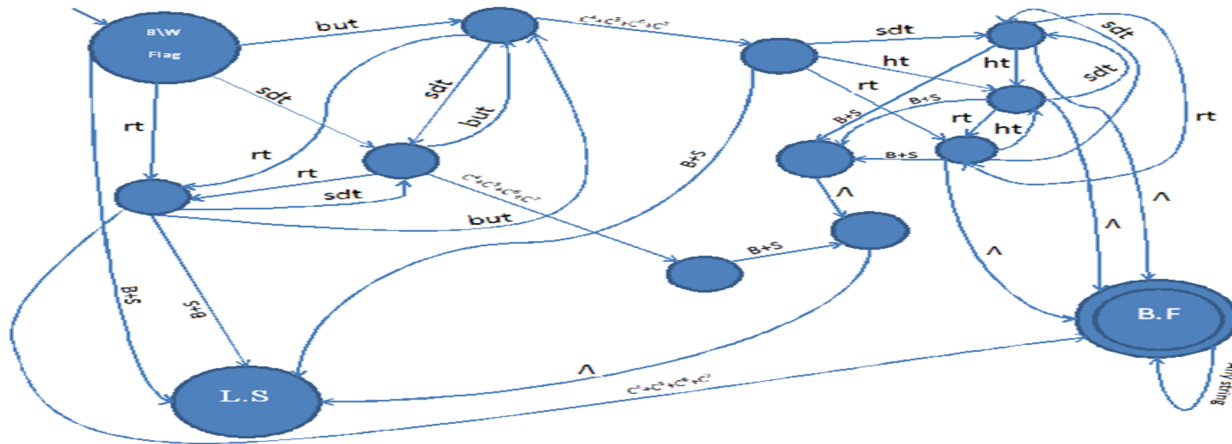


Fig. 12: Automaton showing Roller Coaster detailed design using Transitions with required Σ to lead to win state where win state is $Q(\text{final}) = B.F$.

9) Round Track (rt)

This sigma element will increase the speed of roller-coaster by 17km/h and it will enable the roller-coaster to get the coins which are placed at round shape track. Moreover when a player is at any state and read the input string “rt” the speed of roller coaster will increase and this input alphabet is used when the player have to collect the coins in a circular path or used where roller coaster needs acceleration to reach to the final state.

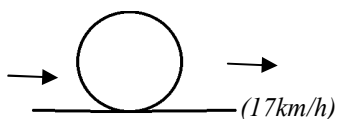


Fig. 13: Roller Coaster collection coins at Round Shape Track at Σrt

10) Energy Capsule (ec)

This sigma element will increase the speed of roller-coaster by 15km/h only on hand make track and it will enable the roller-coaster to get the coins at higher level. Moreover when a certain player is at any state and read the input alphabet “ec” the speed of roller coaster will increase and this input alphabet is used when the player have to collect the coins at a place which is going upward and roller coaster need acceleration. Similarly it is used where roller coaster needs acceleration to reach to the final state and to win the game.

11) Straight Up Track (sup)

This sigma element will increase the speed of roller-coaster by 15km/h and it will enable the roller-coaster to get the coins at vertical path. Also, when a certain player is at any state and read the input alphabet “sup” the speed of roller coaster will increase and this input alphabet is used when a player has to collect the coins at higher place or position which is vertically going upward as shown in fig. 14. It may also be used when roller coaster needs acceleration to reach to the final state when it is at lower position.

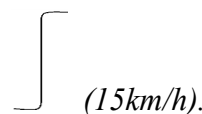


Fig. 15: Coin Collection upward at Σsup

12) Handmade Track (ht)

This sigma element will increase the speed of roller-coaster by 2km/h per 1^0 only in downward direction and it will enable the roller-coaster to get the coins at lower level. However at any state if a certain player reads the input string “ht” the speed of roller coaster will increase or decrease according to the path. That is when roller coaster is used when it needs acceleration or deceleration to reach to the final state.

3- At least four coins must be collected to reach to the final state with a sum of 0\$ reward. Without this must condition, entering into the win state is not possible.

3.1- Collect five coins to reach to the final state with a reward of 100\$. Fifth coin will award the player 100\$, and this is optional we may or may not collect the fifth coin. Thus clause 3.1 may not be counted as a 'pure must condition' and same is the case with 3.2, 3.3 and 3.4. All these clauses though reward the player but are not mandatory.

3.2- If a player collects six coins to reach to the final state, it will be awarded a reward equal to 200\$. Sixth coin will award the player 200\$, and this is optional as a player may or may not collect the sixth coin but still will be able to reach to the final state with four coins.

3.3- A player collecting seven coins will be rewarded 1000\$.

Again, this clause is optional as player will still fall into final state having four coins.

3.4- If a player goes to the final state by crossing the fire ring, it will be awarded 500\$. Crossing from the fire ring is also optional condition as this is only designed for winning more money equal to 500\$.

4- To get to the final state a player must maintain the speed range between 15 km/h up to 40km/h. This clause is to be implemented in the coding phase.

5- At least one stunt is mandatory to reach to the final state and player will be awarded a sum of 1500\$. If user make such a path that will make the roller coaster to be in stunt mode he will be awarded with 1500\$.

6- No sharp edges are allowed, because if there is sharp edge in path roller coaster will break and player will be in lose state.

IV. CONCLUSION

The game is carefully designed using applied automata theory and now is in development phase. It is much easier at development stage if game is initially designed using computational and automata theory tools. It is found that it also has an advantage that bugs are fewer than compared to games designed using any other tool. By designing the automaton of the overall game design and that of certain phases, it has eliminated the need to design cumbersome use case diagrams or using so much of software engineering or design tools at the beginning level of designing the game. A certain regular expression programmed by the DFSA or NFSA covers infinitely many possibilities of input strings to be accepted by the game. It means that all the scenarios which are not possible to be encompassed in the design phase are catered and taken care with by using automata theory applied in game design. Any other tool used will not have the flexibility of deterministic or non-deterministic finite state automata provided by the applied automata theory. If the game is designed using use case diagrams and state transition diagrams from software engineering, though it is efficient at the coding level, however at design stage it will not cover all

possible scenarios a game may follow when it is executed and thus this possibility will lead to more and more effort to be done at the coding stage. Whereas by using regular languages, deploying regular expressions and translating them into automata will cover this broad spectrum of all possible input string accepted by a certain language and that language will be the game in design phase. Hence minimum bugs with maximum efficiency may be obtained by using applied automata theory in design phase in combination with the software engineering best practices at the development stage.

REFERENCES

- [1] C. F. C. C. Anderson, "Experience-weighted Attraction Learning in Sender-receiver Signaling Games," *Economic Theory*, vol. 16, no. 3, pp. 689-718, 2001.
- [2] J. M. J. Andreoni, "Rational Cooperation in the Finitely Repeated Prisoner's Dilemma: Experimental Evidence," *Economic Journal*, vol. 103, pp. 570-585, 1993.
- [3] L. S. Shapley, "A Value for n-Person Games," *Contributions to the Theory of Games*, vol. 2, p. 307-317, 1953.
- [4] M. P. E. Audi Robert, "The many faces of integrity," *Business Ethics Quarterly*, vol. 16, no. 1, pp. 3-21, 2006.
- [5] B. Beizer, *Software Testing Techniques*, vol. second edition, New York: Van Nostrand Reinhold, 1990.
- [6] A. P. G.V. Bachmann, "Protocol testing: review of methods and relevance for software testing," in *Proceedings of the 1994 international symposium on Software testing and analysis*, 1994.
- [7] J. Sifakis, "Modeling real-time systems," in *IEEE Computer Society, 25th IEEE Real-Time Systems Symposium (RTSS '04)*, 2004.