# AutomataMind: A Serious Game Proposal for the Automata Theory Learning

Manuella Vieira, Victor Sarinho

# AutomataMind: A Serious Game Proposal for the Automata Theory Learning

Manuella Vieira[1][0000−0002−3045−4439] and Victor Sarinho[1][0000−0002−5653−8390]

Laboratório de Entretenimento Digital Aplicado (LEnDA)
State University of Feira de Santana, Av. Transnordestina, s/n, Novo Horizonte
Feira de Santana, Bahia, Brazil
manuellavr7@gmail.com, vsarinho@uefs.br

**Abstract.** Digital games facilitate the teaching-learning process, arousing the interest of students in various areas of competence. This paper introduces AutomataMind, a game that seeks to stimulate the player logical thinking skills, while presenting the concepts of Computer Theory with emphasis on Automata Theory in a playful and challenging way.

**Keywords:** Game based learning · Automata theory · Serious games.

## 1 Introduction

Educational software is a significant element in the construction of a knowledge base for students [1]. In addition, digital games have been designed for teaching purposes to engage and motivate the players into learning activities [3].

Studies have shown that many difficulties experienced by students when learning Theory of Computation are related to the lack of logical thinking and problem-solving skills [5], as well as a sense of monotony [6] that is caused by the great number of complex and abstract concepts involved [2].

As a gamified approach to introduce concepts related to the Automata Theory in a playful way, this paper presents "*AutomataMind*", an educational game based on the *Mastermind* board game that seeks to stimulate the player logical reasoning skills through proposed automata challenges.

## 2 Related Work

As an example of digital game related to the Automata Theory learning, *Automata Defense 2.0* [6] is a tower defense game whose objective is to stop a variety of creatures, each one associated with a word, from reaching the map exit. To achieve this goal, the player must design an automaton and assign it to a tower. Each creature that carries a word accepted by a tower automaton is attacked when it moves inside of that tower coverage area. The player is able to position towers in a strategic way, so that they can either attack or slow the creatures movements.

Another game example is the "*Chomsky's Mountain*" [4], where the player goal is to reach the top of a mountain by solving problems that represent the "enemies" that the user must defeat. To do this, the player must build a computing model that matches the theme of the current level of the game. Each level refers to languages associated with two types of grammar from Chomky's Hierarchy: Regular Languages and Context Free Languages.

Regarding the *Mastermind* game, whose game dynamics will be applied in AutomataMind game, it is a code-breaking game that challenges the players by prompting them to guess a specific pattern - also referred to as "the code" - usually composed of a combination of pegs with different colors. Every time the player takes a guess about the code, a feedback is provided showing how close the player's attempt to the actual code was.

## 3   The AutomataMind Game

AutomataMind was designed by integrating Mastermind game play aspects with Automata Theory concepts. The game was structured in the following manner: each level of the game starts with an automaton being shown on the screen. Then, a string - a sequence of characters - that can be accepted by that automaton is generated. This sequence remains hidden and represents the code that should be "broken" by the player.

The player must try to guess what the secret sequence is, based on the automaton that is being presented on the current level screen. Every time the player takes a guess on what the code might be, a certain amount of information regarding the current guess is provided, so that the player next guess can be improved based on that feedback.

There is a limited number of guesses that can be made during a level. Therefore, if the player reaches the limit without having guessed the correct sequence, the stage restarts from the beginning, with a new code that may or may not be similar to the prior one (the sequence selection process is random).
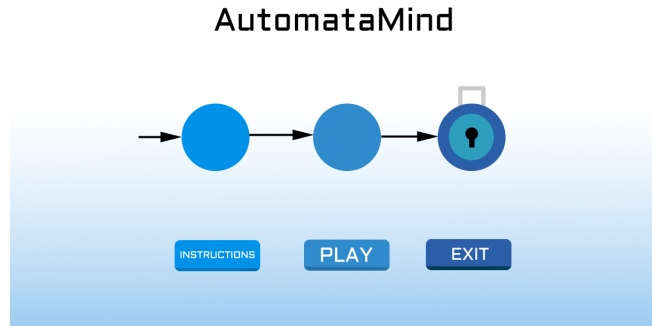


**Fig. 1.** Title screen of the AutomataMind game.

The game main menu (Fig. 1) was designed as a graphic representation of a finite automaton that has a lock as its final state, implying that in order to find the correct code, the player guesses must necessarily be able to reach the automaton accept state. For the menu buttons, three options are provided to the player: *Read* the instructions, *Play* and *Exit* the game.

Fig. 2 illustrates one game level, where the automaton at the top of the screen is used as the player reference to discover the hidden sequence. The pegs in the right side of the screen are used to provide feedback about the player guess, where a **red peg** is placed for each character that is in the same position as a similar character on the secret code. The **white peg** indicates the presence of a matching character - a character that can be found in both the guess and the secret code - but that is placed in the wrong position. The **green confirmation** indicates that the string inserted by the player is accepted by the automaton (meaning it ends in a final state) even though it might not be the correct code.
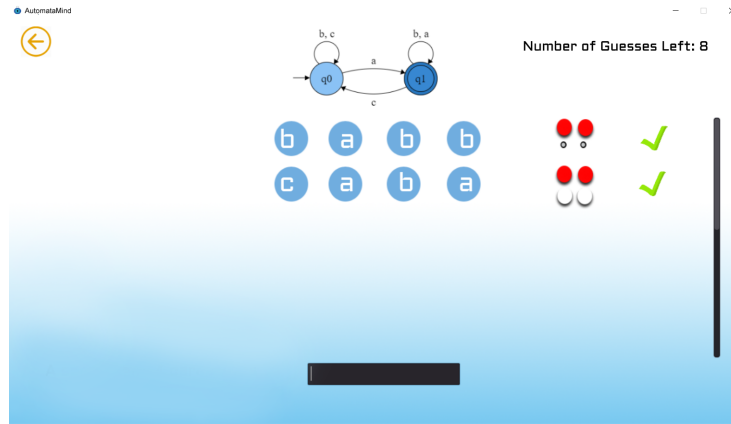


**Fig. 2.** Game play screen of the AutomataMind game.

AutomataMind is also divided into levels with increasing degrees of difficulty, ranging from: number of characters used in the secret code; number of symbols belonging to the automaton alphabet; and number of states and transitions the automaton has. Initially, only the first level of the game is available to the player, and to unlock a specific level, the player must successfully complete the previous one (Fig. 3).

The score mechanism takes into account the number of red pegs the player obtained while playing a level, as well as a multiplier that is inversely proportional to the amount of guesses made. Thus, a greater amount of points is given when the player takes a less amount of guesses to get to the correct sequence.
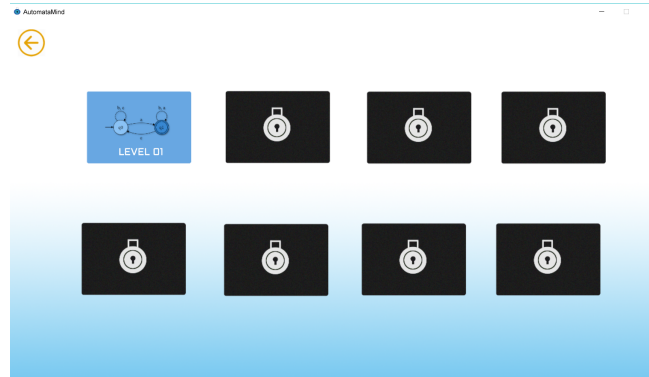
**Fig. 3.** Level selection screen of the AutomataMind game.

## 4    Conclusion and Future Work

This paper presented the AutomataMind, an educational game capable of helping students to develop logical reasoning skills while learning about the Theory of Computation. The developed game is a simple solution that has the possibility of contributing to the scenario of games aimed at teaching topics related to the theoretical aspects of computing.

As AutomataMind is an on-going project (available at **https://github.com/lenda-uefs**), an usability evaluation with undergraduates related to computer science will be performed in the future. The inclusion of Pushdown automata and Turing machines topics in future levels of the game will also be performed for challenging and learning purposes.

## References

1. Bernardi, S.T.: Utilização de softwares educacionais nos processos de alfabetização, de ensino e aprendizagem com uma visão psicopedagógica. Revista REI, Getúlio Vargas **5**(10) (2010)
2. Devedzic, V., Debenham, J.: An intelligent tutoring system for teaching formal languages. In: International Conference on Intelligent Tutoring Systems. pp. 514–523. Springer (1998)
3. Klopfer, E., Osterweil, S., Salen, K.: Moving learning games forward: Obstacles, opportunities & openness: An educational arcade paper (2009)
4. Leite, L., Sibaldo, M., Carvalho, T., Souza, R.: Montanha de chomsky: jogo tutor para auxılio no ensino de teoria da computaç ao. In: XXII Workshop sobre Educaç ao em Informática (WEI 2014), Brasılia, DF (2014)
5. Pillay, N.: Learning difficulties experienced by students in a course on formal languages and automata theory. ACM SIGCSE Bulletin **41**(4), 48–52 (2010)
6. Silva, R.C., Binsfeld, R.L., Carelli, I.M., Watanabe, R.: Automata defense 2.0: reedição de um jogo educacional para apoio em linguagens formais e autômatos. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). vol. 1 (2010)