

RegexMon: An RPG for Teaching Regular Expressions in Computing Education

Lorena Roberta Nunes Guimarães¹, Julio Cezar Costa Furtado¹

¹Departamento de Ciências Exatas e Tecnológicas (DCET)

Universidade Federal do Amapá (UNIFAP)

Rod. Juscelino Kubitschek, KM-02 – 68.903-419 – Macapá – AP – Brasil

lorena.nunes12@hotmail.com, furtado@unifap.br

Abstract. Introduction: Regular expressions are a foundational topic in Computing curricula, yet difficult to teach due to compact notation, limited feedback, and scarce educational games. **Objective:** This paper presents *RegexMon*, an RPG developed in Godot 4 to support teaching regular expressions in undergraduate courses. **Stages:** The game follows a design-based research approach and features a two-phase turn-based battle system: in the attack phase, players construct strings matching a generated regex; in the defense phase, they synthesize a regex that accepts and rejects given string sets, addressing core competencies. Concept progression is organized into six leagues aligned with the typical curriculum. **Expected Results:** *RegexMon* is expected to improve student competence in regex recognition and synthesis, reduce diagnostic difficulties, and increase engagement through open-world exploration and procedural encounters.

Keywords: educational games; regular expressions; formal languages; RPG; game-based learning; computing education.

1. INTRODUCTION

Regular expressions (regex) are a foundational tool in computing, present across formal languages and automata theory, text processing, software validation, compiler construction, and security. Despite this practical relevance, they remain one of the most consistently challenging topics for undergraduate students. The difficulty is threefold: the notation is compact and context-sensitive, with the same symbol carrying different meanings depending on its position; most interpreters provide no clear feedback on the cause of a match failure; and the behavior of a pattern can vary across programming languages and tools, creating an additional barrier for learners encountering regex in real-world contexts [Jargas 2012; Garozi et al. 2021]. The lack of educational materials specific to regex training does not help either. As identified by Garozi et al. (2021), it significantly complicates the student experience of working with regex.

This paper presents *RegexMon*, an open-world RPG game made in Godot 4 that uses monster catching and turn-based fights as the educational mechanism for regex. Taking inspiration from the monster collection games genre, the key gameplay element of the title is its battle system, which consists of two phases corresponding to two regex competencies simultaneously: attack involves recognizing the pattern and building the matching string, whereas defense involves composing the regex based on a number of conditions (acceptance and rejection). Both skills being necessary during each combat means they cannot be bypassed in any way. The curriculum of concepts includes six

leagues of procedurally generated tasks, with one new type of regex constructs introduced during each league.

2. RELATED WORK

This section analyzes three works that share the closest design intent or content scope with RegexMon. Labirinto Gramágico [Garozi et al. 2021] is a digital maze game in Unity in which the player must open doors by constructing regular expressions that match a displayed pattern. Web-based regex practice tools, including Regex Golf, RegexOne, and Regex Crossword, provide drill-based practice environments in which users construct patterns to match or differentiate given string sets. Table 1 summarizes the comparison across six design and pedagogical dimensions.

Table 1. Comparison between related works and RegexMon.

Dimension	Labirinto Gramágico	Regex tools (web)	RegexMon
Medium	Digital (Unity)	Web (browser)	Digital (Godot 4)
Genre	Maze / puzzle	Practice tool	Open-world RPG
Content	Regular expressions	Regular expressions	Regular expressions
Progression	Fixed challenge set	Manual or fixed	6 leagues (procedural)
Regex competence	Synthesis only	Synthesis only	Recognition + Synthesis
Feedback	Match / no match	Match highlight	HP bars + match result
Target audience	CS undergraduates	General / professional	CS undergraduates

3. METHODOLOGY

The development process of RegexMon was based on the method of design-based research where the design of the game was progressively designed taking into account pedagogical principles and feedback on its implementation in three stages including: pedagogical design, iterative implementation, and internal validation. The pedagogical decisions were grounded in established learning theory: the attack and defense duality frames learning as action followed by reflection on its outcome [Kolb 1984], while the graduated progression across leagues introduces one construct at a time within the learner's zone of proximal development [Vygotsky 1978]. Situating the challenges in an authentic debugging-like activity rather than isolated drills draws on situated learning [Lave and Wenger 1991], in line with evidence that game-based approaches support engagement and learning in Computing education [Hosseini et al. 2019].

During the pedagogical design stage, the fundamental elements were determined that reflected basic knowledge about formal languages and especially regular expressions, and further, implemented into gameplay and challenge elements. In the iterative implementation stage, the game was further developed and tested through several prototypes to refine usability, balance and overall interaction experience. The internal validation was conducted throughout development by a specialist professor who accompanied the successive stages and assessed the alignment between the game mechanics and the intended learning goals. The perceptions obtained were favorable regarding the pedagogical adequacy of the two-phase battle design, and the observations raised during this process informed the adjustments made across the iterations.

4. REGEXMON

4.1. Game Concept

The gameplay design of RegexMon is based on the creation of an open-world RPG game that would be played using Godot 4. This game will function as an educational aid during undergraduate Computing courses in order to help students learn more about regular expressions and become better at their use and manipulation. There are many navigable locations on the map, which represent the campus, such as various outdoor locations, interior spaces of buildings, gymnasiums, laboratories, and administrative buildings. The choice of this theme is intentional since it makes the narrative setting relatable to the player. The battles in RegexMon take place in two environments, and the reasons for choosing them have an educational component. Random encounters happen in tall grass areas of the overworld where the player is likely to meet Regexmon. Every time the player moves through this area, there is a probability that the player will face a randomly generated enemy at random league levels. Thus, by walking around and exploring the entire map, the player will face all levels of Regexmon and get ample practice in handling constructs of different complexity.



Figure 1. RegexMon overworld: university campus environment with player character, buildings, and tall grass encounter zones.

4.2. Battle Mechanics, The Attack/Defense Duality

The battle system is RegexMon's central pedagogical mechanism. Each battle consists of two sequential phases that operationalize the two core regex competencies. Figure 2 shows the attack phase interface, and Figure 3 shows the defense phase interface.

Attack phase (recognition). The opponent Regexmon displays a procedurally generated regex according to the current league. The player must type a matching string, validated via a search operation. A correct answer deals 50% damage; an incorrect one reduces the player's HP by one step. After three incorrect responses across both phases, the player is defeated. **Defense phase (synthesis).** The opponent shows three accepted

and three rejected strings generated from a hidden regex. The player must write a regex that accepts all accepted strings and rejects all rejected ones. Validation uses full-string anchoring for precision. A correct answer deals the remaining 50% damage and ends the battle; an incorrect one reduces HP, and the phase repeats with the same strings.



Figure 2. Battle attack phase: the opponent displays a generated regex pattern and the player must type a matching string.



Figure 3. Battle defense phase: the player must synthesize a regex that accepts the listed strings and rejects the others.

The attack phase develops the ability to interpret a regex pattern and reason about the strings it accepts, a competency required when reading validation rules, parsing existing code, or debugging a pattern that produces unexpected matches. The defense phase develops the ability to construct a regex from behavioral specifications, the

competency required when writing input validators, search patterns, or format constraints from scratch. Because both are required in every battle, neither can be practiced in isolation or avoided through play choices.

4.3. Procedural Progression by League

The concept curriculum is organized into six leagues, each league introduces a new level of regular expression construct and at the same time keeps all the constructs introduced in the preceding leagues. At each battle, the league of the battle is randomly assigned, thus exposing the user to all the levels of difficulty. Table 2 summarizes the progression.

Table 2. RegexMon league progression and regex constructs introduced per league.

League	Regex constructs introduced	Example patterns generated
1	Literals (lowercase, uppercase)	ab, XYZ, mNp
2	+ Character classes: ranges [a-z], [A-Z], [0-9]; negation [^]	[a-f]bc, [^aeiou][0-9]
3	+ Quantifiers: *, +, ?, {n}, {n,m}	[a-c]+, x{2}, ab?c
4	+ Anchors: ^, \$, \b	^[A-Z][a-z]+\$, \bword\b
5	+ Groups and alternation: (...),	(ab cd)+, (foo bar)\$
6	+ Escape sequences: \d, \w, \s, \D, \W, \S; wildcard .	\d{3}-\w+, [a-z]+\s\D

4.4. Feedback System

RegexMon's feedback operates on two levels. The primary level is the match result itself: the player's string either matches the opponent's regex or it does not, and this outcome is communicated immediately through the battle dialogue and HP animation. There are no error messages, syntax warnings, or explanatory text, the player must infer the cause of a failure by examining the relationship between the submitted string and the displayed pattern, performing the same diagnostic reasoning required in professional regex debugging. This is what Pérez et al. (2018) classify as intrinsic feedback: the consequence of the decision is embedded in the task environment itself. The secondary level is progression feedback, which operates across the encounter rather than within a single attempt: the depletion of the HP bars over the course of a battle, the victory or defeat outcome reached after the allowed number of attempts, and the advancement through the six leagues together signal the player's accumulated performance and indicate which constructs still require practice.

5. CONCLUSIONS AND FUTURE WORK

This paper presented RegexMon, an open-world RPG built in Godot 4 for teaching regular expressions in undergraduate Computing courses. Its central contribution is a two-phase battle system that operationalizes both regex competencies, recognition and synthesis, so neither can be avoided through play. Progression spans six leagues of procedurally generated challenges of increasing complexity. The prototype is fully functional, with both battle phases, procedural generation, and ten original creatures.

A controlled experiment is planned to evaluate learning gains with the IAQJED instrument, using pre- and post-tests on regex recognition and synthesis with undergraduates.

REFERENCES

- Garozi, H.; von Wangenheim, C. G. and Hauck, J. C. R. (2021). Labirinto Gramágico: Um Jogo Educativo para o Ensino de Gramáticas Regulares. In: *Anais do XX Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2021)*.
- Hosseini, H.; Hartt, M. and Mostafapour, M. (2019). Learning is child's play: game-based learning in computer science education. *ACM Transactions on Computing Education*, v. 19, n. 3, p. 1–18.
- Jargas, A. M. (2012). *Expressões Regulares: Uma Abordagem Divertida*. 3. ed. São Paulo: Novatec.
- Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice-Hall.
- Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.
- Pérez, E. del M.; Duque, A. P. G. and García, L. C. F. (2018). Game-based learning: increasing the logical-mathematical, naturalistic, and linguistic learning levels of primary school students. *Journal of New Approaches in Educational Research*, v. 7, n. 1, p. 31–39.
- Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Cambridge: Harvard University Press.