Interactive Cybersecurity Awareness: Creating a Gamified Password Strength Checker with Unity

Miloš Kostić[1,*], Igor Saveljić[1,2]

[1]Faculty of Information Technology, Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade, Serbia
[2]Institute for Information Technologies, University of Kragujevac, Jovana Cvijića bb, 34000 Kragujevac, Serbia

Abstract

In an increasingly digital world, passwords serve as crucial barriers protecting personal information across various online platforms. However, many users remain unaware of the significance of strong password practices, making them vulnerable to cyber threats. This paper explores the urgent need for enhanced security awareness and improved password strategies, proposing the integration of gamification as an innovative educational approach. Paper discusses the development of the "Lockedout" game, which employs a Unity-based password strength checker that not only evaluates passwords against traditional criteria but also incorporates advanced techniques, including Markov Models. These models analyze character transitions to provide a more nuanced assessment of password strength, particularly against common vulnerabilities. By transforming the learning process into an engaging, interactive experience, "Lockedout" aims to foster better password habits among players, making cybersecurity education both effective and enjoyable. Ultimately, this paper illustrates how gamification can significantly impact the understanding and implementation of strong password practices, addressing a critical gap in current cybersecurity efforts.

Keywords

Gamification, Cybersecurity awareness, Unity, Password strength validation, Markov Model

1. Introduction

In an era where our lives are increasingly intertwined with digital platforms, passwords have emerged as a vital line of defense in protecting our personal information. From social media accounts to online banking, passwords serve as the gatekeepers to our digital identities. However, many users remain unaware of the critical importance of strong passwords, leaving them susceptible to cyber threats. This paper aims to tackle this pressing issue by enhancing security awareness and encouraging better password practices through the innovative use of gamification.

This paper will delve into the significance of raising awareness around password strength and explore how gamification can be effectively integrated into educational frameworks. Additionally, it will present the ongoing development of the "Lockedout" game, specifically focusing on the Unity implementation of password strength checkers [1]. We will analyze potential weaknesses in standard password checking algorithms and address them through the implementation of a Markov Model approach, tailored for more advanced players. By merging entertainment with education, this paper seeks to illustrate how gamification can create a more impactful learning experience in the realm of password security.

2. Importance of password strength awareness

The rapid expansion of online services has made secure authentication mechanisms more important than ever. At the center of this security are passwords, which are the primary method of protecting personal and sensitive data from unauthorized access. Weak passwords remain one of the most exploited vulnerabilities in cyberattacks [2], often providing an easy entry point for attackers through brute force,

BISEC'2024: 15th International Conference on Business Information Security, November 28-29, 2024, Niš, Serbia
*Corresponding author.
milos.kostic@metropolitan.ac.rs (M. Kostić); igor.saveljic@metropolitan.ac.rs (I. Saveljić)
0009-0005-0912-9518 (M. Kostić); 0000-0002-0707-5174 (I. Saveljić)

dictionary attacks, or credential stuffing [3]. Despite technological advances, many users still rely on simple, predictable passwords, leaving their accounts at risk.

Weak passwords often result from users prioritizing convenience over security. Simple passwords are easier to remember but equally easy to guess. With data breaches becoming more common, weak passwords are frequently found on password blacklists, where they are cataloged and used for automated attacks. The growing sophistication of cyberattacks makes it vital for users to create passwords that resist such tactics.

Creating and managing strong passwords remains a significant challenge for users, prompting a wealth of research dedicated to this issue [4, 5]. Recent studies have shown that some password meters do effectively guide users toward better password choices, offering a glimmer of hope in this area. However, these meters are typically based on ad hoc designs, and most vendors fail to provide transparency regarding their design choices [6].

Traditional checkers often rely on basic rules, such as requiring specific numbers or special characters, resulting in low accuracy, where insecure passwords may be accepted while secure ones are rejected. This can harm both security and usability, as users frequently resort to predictable modifications (e.g. "password1") [7]. Adaptive Password Strength Meters based on Markov models [7, 8] appeared as a valid solution to this problem, since they provide a more accurate assessment of password strength by estimating the probabilities of the n-grams that comprise the passwords.

Markov Models are commonly used in statistical prediction and pattern recognition and can offer more sophisticated password strength evaluations by analyzing character sequences and predicting the likelihood of certain character combinations. In password security, a Markov Model helps estimate the strength of a password based on the probability of its character sequence appearing in common datasets, such as often online available dictionaries of known passwords.

In light of the above, it is essential to enhance awareness of password security to address one of the most significant gaps in cybersecurity today. Many users do not realize how vulnerable their accounts are to compromise through weak or reused passwords.

3. Gamification

Gamification has grown into a powerful tool across various industries, from education to marketing [9], and most notably, in enhancing user engagement with otherwise mundane tasks. In this context, gamification refers to the incorporation of game mechanics, such as points, challenges, and rewards, into non-game activities to encourage participation and motivation. By turning traditionally often dry subjects like cybersecurity into an interactive experience, users are more likely to retain information and develop better habits.

Gamification is particularly relevant in the field of cybersecurity awareness [10, 11, 12] because it transforms learning about security practices, such as importance of creating strong passwords, from a chore into a challenge. By integrating these elements into a game form, players not only have fun but are also subconsciously learning valuable skills. The mechanics of password creation, strength evaluation, and cracking are turned into engaging, playable scenarios that reinforce good security habits. Through this approach, the game promotes active learning and critical thinking, making players more aware of the risks associated with weak passwords and the importance of creating strong ones.

Research has shown that gamified learning increases retention and motivates users to engage more deeply with the material [13]. In cybersecurity, where awareness is crucial, this method ensures that users internalize the importance of strong password practices. The "Lockedout" game's core mechanic [1], evaluating password strength, plays directly into this by challenging players to think critically about password creation. The end result is a player base that is not only entertained but also better prepared to handle real-world security challenges.

Figure 1: Current game title/logo design.

Figure 2: Desktop.

4. ”Lockedout” Game concept

”Lockedout” [1] aims to blend education with entertainment, and it is designed to teach players about password strength and cybersecurity best practices (Fig. 1.) through some interesting time based challenges. Game introduces players to the fundamentals of password strength in an intuitive and engaging way, raising awareness in a manner that goes beyond typical informational campaigns. Design emphasizes the significance of strong passwords and encourages users to adopt better habits especially when making decisions in time limited situations.

Game concept considers overall slow start with an old computer monitor featuring an operating system login window. Scattered sticky notes with careless credentials invite players to access the system. Upon entry, they encounter a desktop with a text document titled "MyPasswords.txt" and five applications: "Email," "Bank," "Chat," "Social," and "Shopping" (Fig. 2.).

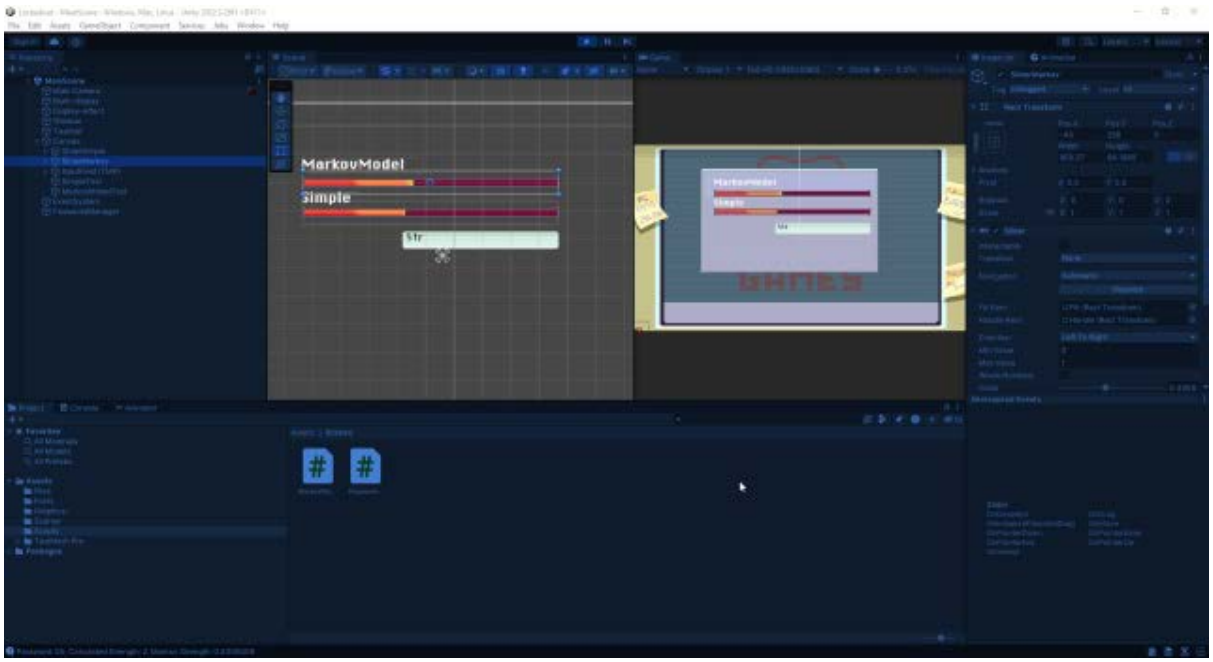

Figure 3: Unity development environment.

As players interact with the chat application, a sudden notification from a friend about recent data breaches interrupts their conversation, accompanied by an OS Guard (some form of system cybersecurity protector) warning of an ongoing cyberattack. This pushes players into a race against time to replace passwords for personal profiles in every app. Each app features a red timer that emphasizes the urgency of changing passwords. Players must meet strict password strength criteria while managing limited change attempts. Failure to act leads to hackers taking control of their accounts, intensifying the urgency. The ultimate goal is to secure as many accounts as possible, culminating in a victory for those who succeed.

After the game, an epilogue summarizes password security best practices and explains the weaknesses of the initial passwords, providing players with resources for further learning.

5. Unity Engine

Unity [14] is one of the most widely used game engines in the world, known for its versatility and cross-platform capabilities. It provides developers with the tools necessary to create 2D and 3D games, simulations, and other interactive content. Unity's ease of use, combined with its powerful scripting engine based on C#.

The decision to use Unity for "Lockedout" was primarily based on the engine's flexibility and strong community support. Unity's intuitive development environment allows for rapid prototyping, which is essential for a game that needs constant testing and refinement. For this project, Unity's scripting flexibility made it easy to integrate the password checker algorithm, enabling real-time feedback for players.

Unity's robust C# scripting environment was particularly advantageous in building the password strength checker. By leveraging Unity's UI and backend functionality, the password checker seamlessly integrates into the gameplay, providing real-time strength feedback and enhancing the overall player experience.

6. Implementation

The password strength checker, along with OS Guard is integral to the game's progression system. As players attempt to solve in-game puzzles, they must generate passwords that pass the strength requirements which will depend on game difficulty level.

Basic idea is that at low difficulty level, aimed for players with basic or no cybersecurity knowledge, next to lower strength value required to pass, a traditional password checker evaluates the length, use of numbers, capital letters, and symbols. However, at higher difficulty levels, the game will introduce a more advanced checker, incorporating a Markov Model, which prevents the use of weak, potentially blacklisted passwords.

This provides an additional layer of challenge, ensuring that the player is constantly encouraged to think critically about password security.

6.1. Traditional Password Strength Checker

The first step in the development of the password strength checker for the "Lockedout" game was the implementation of a traditional password checking algorithm. This foundational system is designed to evaluate passwords based on standard criteria that are commonly used in most password validation systems. These criteria include:

- Password length: Ensuring the password meets a minimum number of characters (often 8 characters).
- Use of uppercase letters: Encouraging the inclusion of at least one uppercase letter to add complexity.
- Use of lowercase letters: This helps prevent the use of all-uppercase passwords, which can also occur accidentally if the "Caps Lock" feature is enabled while creating the password.
- Use of numbers: Checking for the presence of numerical digits within the password.
- Inclusion of symbols: Ensuring special characters or symbols (e.g. ! " # $ % & ' ( ) – 33 characters in total) are used to further increase password difficulty.

Traditional checker is implemented in C# script by using simple if/else if conditions and integrated directly into the Unity engine. For each condition password successfully meet, strength is rewarded with added value. Total score password can achieve goes from 0 to 5. The system automatically validates each password while it is being entered by the player and assigns a corresponding strength level based on the criteria mentioned above.

The engine's ease of handling UI elements, such as text input fields and slider elements provided the perfect platform to implement this feature smoothly. The checker updates immediately while player enter passwords, adjusting the strength meter slider shown on screen, offering immediate feedback to the player.

However, while this system covers the basics and introduces players to the concept of password strength, it is not foolproof. One significant limitation is its inability to identify blacklisted or commonly used passwords. This shortfall opens up the possibility that easily guessed or widely used passwords (e.g., "123456" or "password") could still pass as valid. This limitation is especially concerning in the context of the "Lockedout" goal of educating players to properly define passwords.

6.2. Markov Model Probability Integration

To address the shortcomings of the traditional password checker and provide an additional layer of complexity which can be used in the hard difficulty of the game, two modifications are introduced:

- Detection of black listed passwords
- Markov Model of probability calculation

For sake of providing working data for both mentioned changes, at the beginning of the game a dataset of 999998 known weak or blacklisted passwords obtained in form of the text document [15] is loaded.

First problem of eliminating blacklisted password as a valid option is done through simple comparison of entered password with created and loaded list of passwords, while also notifying the player about the match.

The basic idea of Markov Model implementation involves training the model on a large dataset of real-world passwords to learn the probabilities of transitions between characters or groups of characters (n-grams). Once trained, the model can estimate the probability of successfully breaking a password through cyber-attacks.

During the training process, model might learn for example that the letter "e" often follows the letter "th" (as in "the") or those certain letter combinations like "the" occur more frequently.

In the context of the “Lockedout” game, third-order Markov Model was chosen for the implementation, which considers the probability of a character appearing based on the three preceding characters. This allows more refined estimation of the password’s structure and can flag weak or predictable passwords that the traditional checker might miss. Some initial tests were done with first order Markov Model but they gave poor results, because player could get false positives. The model is fundamentally composed of two dictionaries: one that tracks transition counts between pairs of characters (mapping the current pair to the next character) and another that records the total occurrences for each character pair.

Model is trained on the list of nearly 1 million most often used passwords. Training process considers iteration through each character in the password player enters, by observing groups of three characters. Algorithm, initially selects the first two characters as a pair, followed by the third character that comes after this pair. For each pair of characters, the frequency of a specific third character appearing after them is tracked. If the pair of characters has not been encountered before, a space in memory is allocated to store the potential characters that could follow. Each time the same pair of characters is followed by the same third character, the count is increased. This process builds a map of how pairs of characters transition to others, capturing common patterns in the passwords and creating a record of the most likely sequences.

The model then assigns a probability score to any entered password, and the high score indicates a likelihood of the password being weak or easily guessable.

Implemented Markov Model algorithm required additional adjustments in order to reduce the penalty added to probability estimation which by default occurs as password length increase. The resulting probability estimation is combined with the basic algorithm’s strength estimation to produce a final result. These values can be integrated in various ways. In this case, we decided to subtract the probability from the strength estimation to lower the score for highly probable passwords while rewarding unique ones. It is important to note that some tests were conducted with equal contributions from both values (50% each), but the results were not satisfactory for our purposes.

Goal of introducing the Markov Model was to make the password strength checker more robust. Idea that game system can differentiate between a password that merely satisfies basic criteria (length, symbols, numbers, etc.) and one that is truly secure was valuable considering that game will have different difficulties.

6.3. Testing and comparison

Various tests were conducted on the latest version of the code to evaluate the performance of each algorithm, and the results are presented in three different sliders within the game for testing purposes.

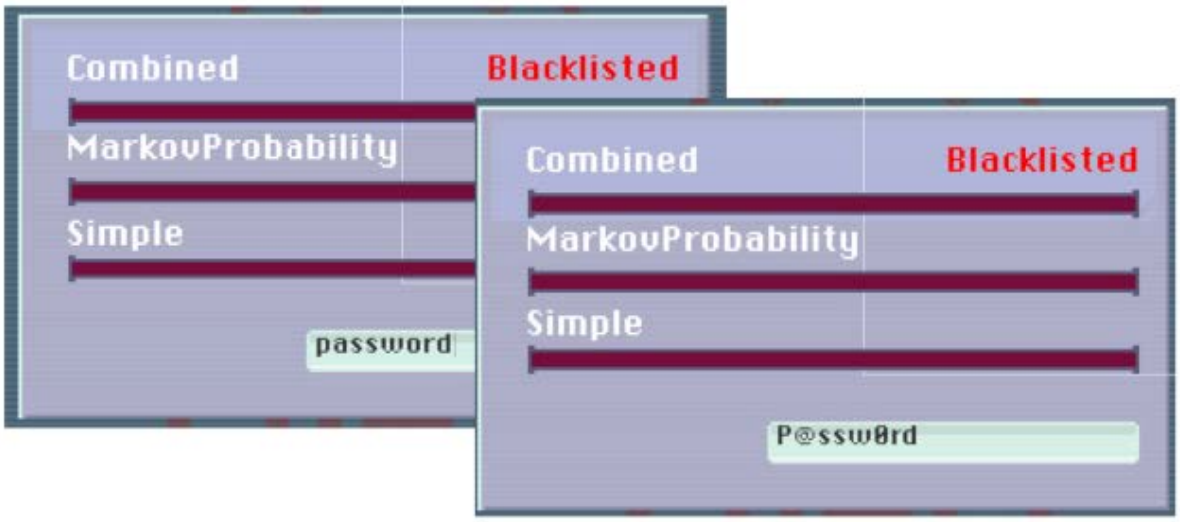

Figure 4: "password" and "P@ssw0rd" blacklisted tests.

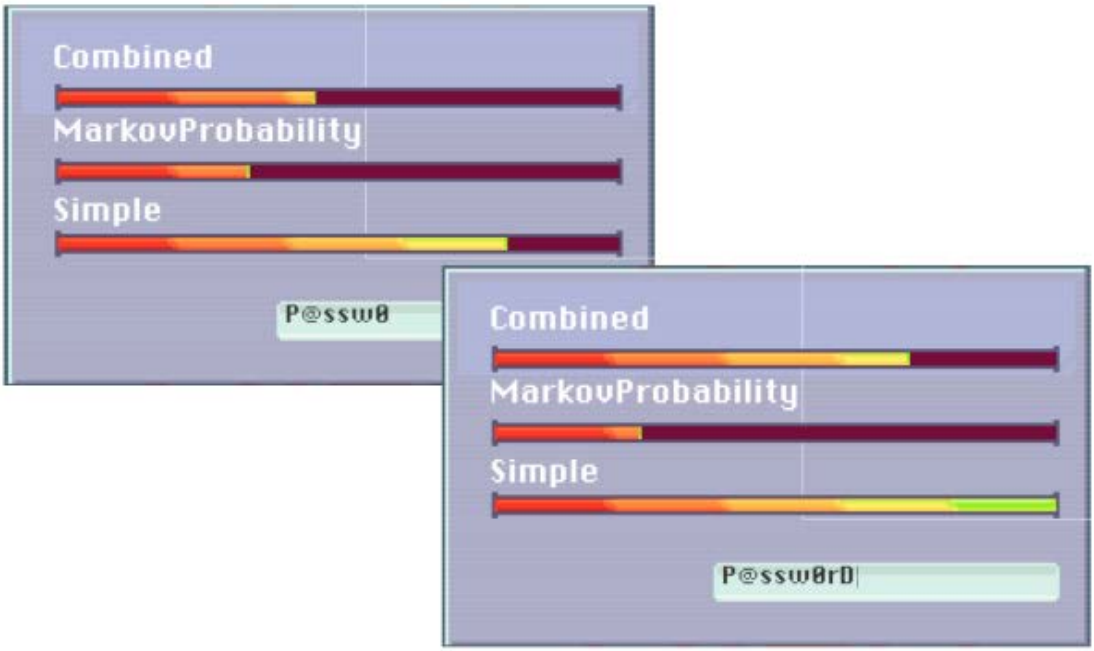

Figure 5: "P@ssw0r" and "P@ssw0rD" tests.

1. Several tests with known blacklisted passwords were positive and screen notification of blacklisted password was activated (Fig. 4.).
2. Modifications of the entry "P@ssw0rd" gave some results, but it's clear that combined approach better detects still problematic entry (Fig. 5.).
3. Further changes by adding special characters to distance from previous entries show improvement in results (Fig. 6.).
4. Last test was done with automatically generated password, using Avast online password generator [16]. As expected this kind of password assembled through random combination of letters, numbers and signs achieved high score, which is why these kinds of passwords are often used by various high profile password manager applications. (Figs. 7. and 8.)

The numerical results of the tests are presented in Table 1, where T denotes the Traditional algorithm,

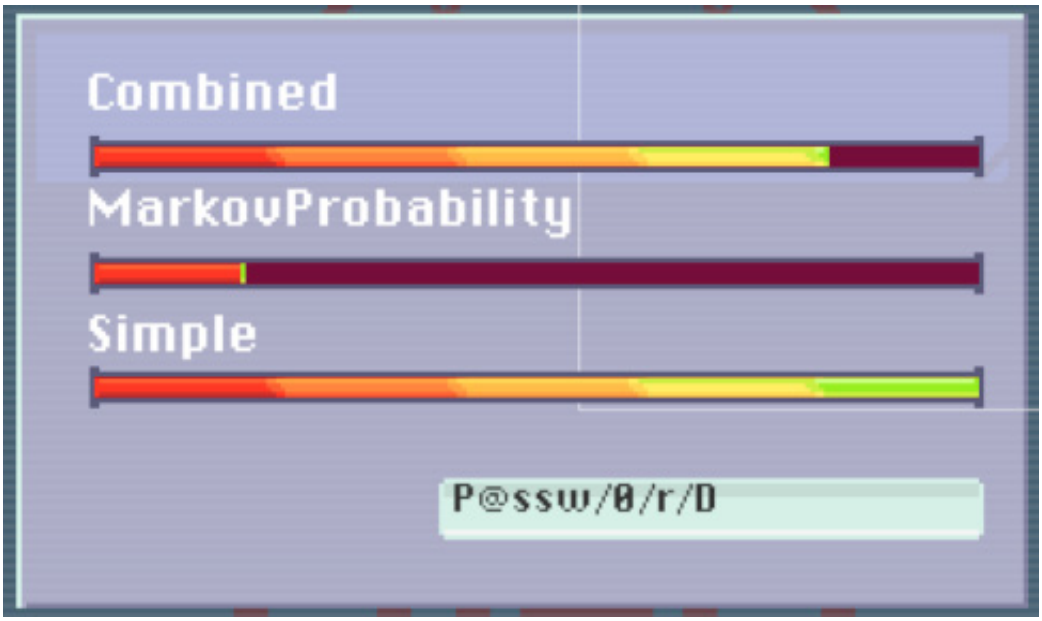

Figure 6: "P@ssw/0/r/D" test.

Figure 7: Avast online password generator [16].

M represents the Markov Model probability value, and C indicates the combined algorithm.

In terms of computational efficiency, the basic checker is much faster as it involves only a handful of straightforward conditions. The Markov Model, while more computationally intensive due to its need to calculate probabilities for sequences of characters, still operates efficiently enough to be used in real-time gameplay within Unity. No visible delays were noticed.

From the perspective of player experience, the basic checker represents easier challenge and provides

Table 1
Numerical results of performed password tests.

Password	T	M	C
P@ssw0	4	1.706046	2.293954
P@ssw0rD	5	1.306699	3.693301
P@ssw/0/r/D	5	0.849964	4.150036
o&7!MclVY9FaZY=	5	0.1710646	4.828935

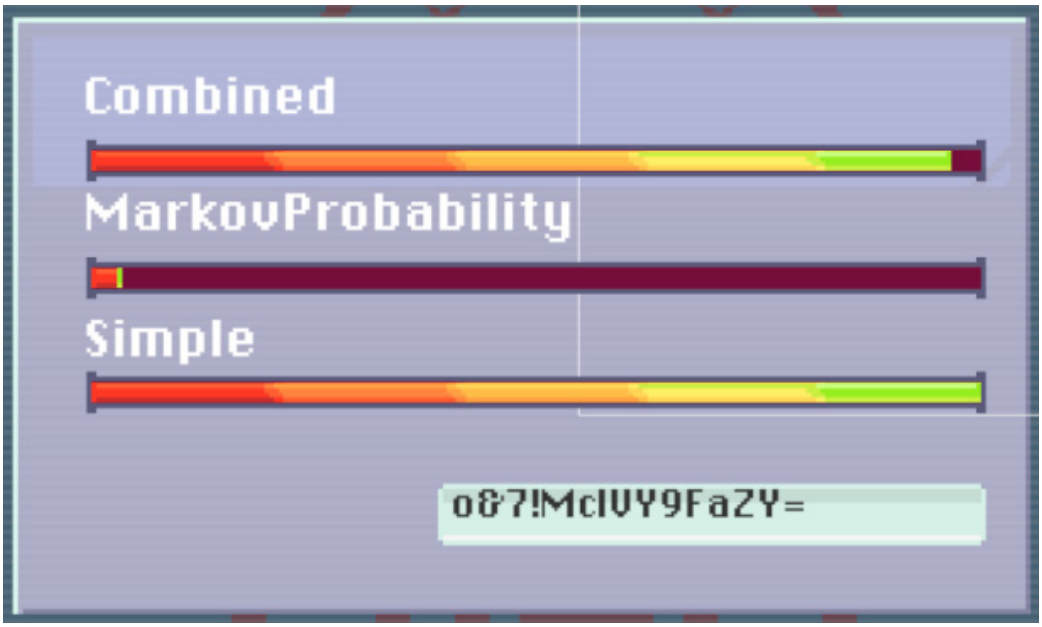

Figure 8: “o&7!MclVY9FaZY=” test.

immediate and clear feedback, making it suitable for introductory or easier difficulty levels. However, as players progress into the harder difficulties, the Markov Model brings in a new layer of challenge. Players must think more critically about the passwords they create, contributing to both the educational and gaming aspects of ”Lockedout”.

7. Conclusion and Future Work

The integration of a password strength checker within the ”Lockedout” game introduces a novel approach to enhancing security awareness through gamification. Password security is a critical aspect of digital safety, yet traditional educational methods often lack the engagement needed to make a lasting impact. By incorporating both a basic password validation algorithm and an advanced third-order Markov Model, this project demonstrates how gamification can overcome these limitations, offering a more engaging and effective learning experience.

The traditional password checker in ”Lockedout” serves as a key tool for educating players on the core principles of secure password creation, ensuring an interactive introduction to basic security concepts. However, the inclusion of the Markov Model and blacklisted password checking adds a significant layer of sophistication, enabling the detection of common patterns and sequences that basic algorithms may miss. This progression not only heightens the challenge within the game but also reinforces the necessity of unpredictable and secure passwords in real-world applications.

Looking to the future, potential improvements will include the incorporation of OS Guard AI agent which will be able to provide real-time instructions, notifications and suggestions to the player. Password strength checkers are just part of the mechanism, and without appropriate educative prompts and tips gameplay would not be complete.

Additionally, mechanic which summarize player’s actions and explains good and bad practices in cyber-security could further enhance the game’s educational value and overall user experience.

Acknowledgment

This paper was supported in part by the Video Game Development Laboratory and in part by the Blockchain Technology Laboratory, both at Belgrade Metropolitan University, Belgrade, Serbia.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Kostić, I. Saveljić, Gamification as a tool for elevating password strength awareness, in: Proceedings of the Fourteenth International Conference on Business Information Security BISEC'2023, Niš, 24th December 2023., 2024, pp. 18–22.

[2] L. A. Shepherd, J. Archibald, R. I. Ferguson, Perception of risky security behaviour by users: Survey of current approaches, in: Human Aspects of Information Security, Privacy, and Trust: First International Conference, HAS 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013. Proceedings 1, Springer, 2013, pp. 176–185.

[3] G. Hu, On password strength: a survey and analysis, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (2018) 165–186.

[4] S. L. Pfleeger, D. D. Caputo, Leveraging behavioral science to mitigate cyber security risk, Computers & security 31 (2012) 597–611.

[5] J. M. Stanton, K. R. Stam, P. Mastrangelo, J. Jolton, Analysis of end user security behaviors, Computers & security 24 (2005) 124–133.

[6] X. D. C. D. Carnavalet, M. Mannan, A large-scale evaluation of high-impact password strength meters, ACM Transactions on Information and System Security (TISSEC) 18 (2015) 1–32.

[7] C. Castelluccia, M. Dürmuth, D. Perito, Adaptive password-strength meters from markov models., in: NDSS, 2012, pp. 1–14.

[8] V. Taneski, M. Kompara, M. Heričko, B. Brumen, Strength analysis of real-life passwords using markov models, Applied Sciences 11 (2021) 9406.

[9] P. Herzig, M. Ameling, B. Wolf, A. Schill, Implementing gamification: requirements and gamification platforms, Gamification in education and business (2015) 431–450.

[10] S. Scholefield, L. A. Shepherd, Gamification techniques for raising cyber security awareness, in: HCI for Cybersecurity, Privacy and Trust: First International Conference, HCI-CPT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings 21, Springer, 2019, pp. 191–203.

[11] G. Fink, D. Best, D. Manz, V. Popovsky, B. Endicott-Popovsky, Gamification for measuring cyber security situational awareness, in: Foundations of Augmented Cognition: 7th International Conference, AC 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013. Proceedings 7, Springer, 2013, pp. 656–665.

[12] I. Rieff, Systematically applying gamification to cyber security awareness trainings: A framework and case study approach, 2018.

[13] G. Barata, S. Gama, J. Jorge, D. Gonçalves, Improving participation and learning with gamification, in: Proceedings of the First International Conference on gameful design, research, and applications, 2013, pp. 10–17.

[14] Unitiy Technologies, 2024. URL: https://unity.com/.

[15] D. Miessler, 10 million password list top 1000000, 2020. URL: https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt.

[16] Avast, Online random password generator, 2024. URL: https://www.avast.com/random-password-generator#pc.