The Royal Crush: Analysis of Match-3 Mechanics

Daniel Eckmann
Games Engineering
University of Würzburg
Würzburg, Germany

Kai Schiesser
Games Engineering
University of Würzburg
Würzburg, Germany

Sebastian von Mammen
Games Engineering
University of Würzburg
Würzburg, Germany

***Abstract*—Match-3 games are a hugely popular genre of video games, with games like Candy Crush or Royal Match getting millions of downloads and generating revenues of multiple million U.S. dollars per month. These games come in many different forms, with different playstyles and core mechanics, but all share the basic goal of matching pieces to make them disappear. In this paper, a toolkit for the creation of Match-3 games using the Godot Engine is presented, implementing a variety of common mechanics found in many such games. This toolkit was used to conduct a study on the difficulty arising from the deployment of different mechanics. The results are presented and discussed to gain insights into the level design tendencies of Match-3 Games. It is found that games using the swapping control type are generally more difficult than games that use the collapse control type, as well as that the number of special pieces used, influences the difficulty, and a lower amount of special pieces should be used in earlier levels to properly balance the difficulty of these games.**

***Index Terms*—Match-3, game mechanics, difficulty, flow**

I. Introduction

Match-3 games are a popular genre of video games worldwide, with games like Candy Crush and Royal Match attracting millions of players and achieving annual revenues of more than a billion dollars [1]–[3]. Their short levels and non-committal gameplay make them suitable for playing on the go or whenever there is a little downtime. In this paper, we first present an open-source toolkit [1] that allows for quick composition of Match-3 games. Second, we present a study, that the toolkit allowed us to conduct, on the difficulty arising from the deployment of different Match-3 mechanics and their combinations.

II. Related Work

The genre title Match-3 exposes the games' main mechanics of matching three or more board pieces [4], [5]. Despite the many variants of the underlying rules, Match-3 games are commonly easy to understand and easy to play, typically in quickly succeeding levels [6], [7]. Colourful animations and effects provide a sense of accomplishment with every match made, and the variety in level scenarios and continuously changing configurations keep the player challenged.

By evaluating the effects of different board configurations on the players' performance, we can measure the experienced difficulty [8]. The results of this analysis can provide a basis for designing levels that neither demand too little nor too much from the player, thus ensuring an engaging "flow" experience [9]. Maintaining flow has, in fact, been identified as the main challenge in keeping players engaged in the popular Match-3 game Candy Crush Saga [10]. A systematic analysis of Match-3 level designs has been published in [11]. Especially the increasing level complexity found in state-of-the-art games guided the exploratory analysis to link mechanics and difficulty presented in this paper.

III. Match-3 Toolkit

To support our research, we developed a Match-3 toolkit for free, open-source game engine Godot [12]. We integrated one mechanic at a time, to arrive at a comprehensive, interwoven mechanics set. We generalized its implementation and devised a clean API that allows for the concrete specification of a game. The basic game loop consists of the player making a move, followed by searching and detecting any matches that have been made on the board, relying on an adapted version of the flood fill algorithm [13]. A simple implementation of a match implies three or more pieces of the same type placed in a vertical or horizontal line. Matching pieces receive damage and if their health values drop to 0, they are removed from the board. Depending on the configuration of the matched cluster and based on the specified rules of the game instance, special items such as bombs might be spawned at the gaps created. The remaining gaps are typically filled by random pieces dropping from above.

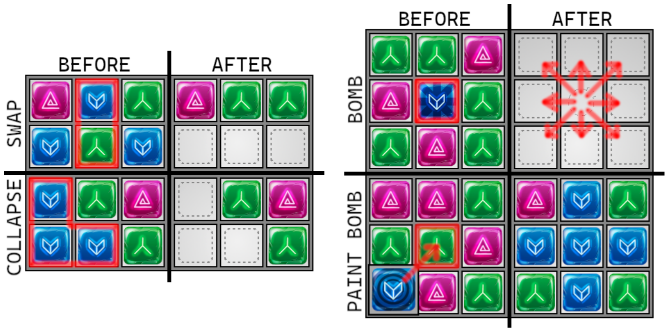
We implemented the most basic set of common Match-3 mechanics: For **core mechanics**, we implemented *swapping* (later denoted as $\mathcal{s}$) and *collapse* (c) as shown in Fig. 1(a), for **gadgets**, we implemented *bombs* (bmb), *jelly fish* (jf), and *paint bombs* (pb) (Fig. 1)(b)). As **obstacles**, we provide *blockers* (blk), *locked elements* (lck), *removable obstacles* (rem), *movable obstacles* (mov), *growing obstacles* (grw), *time bombs* (tmb), and *shielded pieces* (shd) (Fig. 2). The toolkit currently allows for specifying three different **goals**: *Clearing* a level (clr), *clearing specific* subsets, e.g. removing a certain number of pieces of a specific color, and playing for a high *score* (sc).

IV. Measuring Difficulty

The toolkit supports the collection of certain metrics during gameplay. They include the numbers of **wins** and overall

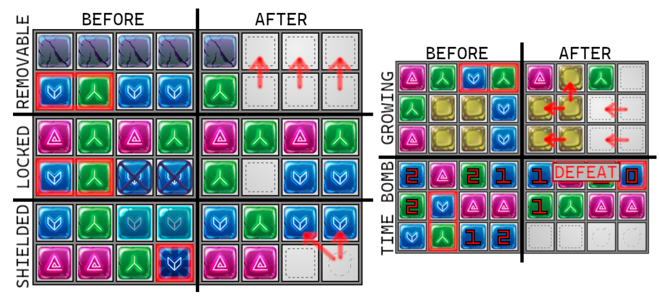
[1] https://github.com/DanielEckmann/Match3Toolkit

979-8-3503-5067-8/24/$31.00 ©2024 IEEE



(a) (b)

Fig. 1: (a) Top: The red-framed pieces are swapped, creating a blue row which is removed. Bottom: The player selects the cluster of 3 blue pieces to let it collapse. (b) Top: An exploding bomb removes a 3x3 square. Bottom: Dropping a paint bomb changes the types of a cluster of pieces.



(a) (b)

Fig. 2: (a) A match removes neighbouring obstacles. Centre: A match unlocks two pieces. Bottom: An explosion removes the shields of two pieces. (b) Top: A match removes growing obstacles (in yellow), the remaining ones spread. Bottom: A match defuses time bombs in the bottom row (with numbers), but one in the top row still ignites and game ends.

trials of a specific configuration, the **time** needed to clear a level, the **time** between moves, the **moves** needed to clear the level, and the achieved **score**. To facilitate the evaluation, they are automatically tracked and saved upon level completion. Aligned with [8], we chose metrics that can be observed, quantified independently of a concrete level configuration, and are directly influenced by the gameplay. To this end, we developed a set of functions that use the collected data to arrive at a quantifiable difficulty score D, see Eqn. 1.

$$D = \frac{1}{n}\sum_{n}\frac{t}{t_{max}} + \bar{t}_m + \frac{n-s}{n} \tag{1}$$

It calculates the difficulty score for a specific scenario. Playing the scenario n times, the first term calculates the average normalised play time, as each trial takes at most $t_{max} = 120s$. $\bar{t}_m$ represents the mean time between individual moves across all n trials, which has been min-max normalized over the whole data set across all scenarios. Finally, as s denotes the number of won trials, the last term represents the overall "unsuccess rate" for c of unsuccessful versus total trials. Every term yields a value between 0 and 1, and all terms are weighted equally. Longer times needed for completion, a lower frequencies of moves, and lower success rates yield higher difficulty scores. D can assign difficulty scores to all configurations with the objective of fully clearing a level or clearing specific subsets of pieces. There are also scenarios which cannot be failed, which run for a set amount of time, and thus merely concentrate on maximising the players' scores. Analogous to Eqn. 1 we designed a difficulty score D_{sc} for these scenarios in Eqn. 2.

$$D_{sc} = \bar{t}_m + (1 - \frac{sc}{sc_{max}}) \tag{2}$$

In addition to $\bar{t}_m$, D_{sc} considers the normalized achieved score, whereas sc is the measured score and sc_{max} is the highest possible score for the given scenario. For a lack of an optimal strategy of play at this point, we approximated sc_{max} by the highest achieved score in our study. In total, D_{sc} increases with lower achieved scores and with higher mean times between moves.

V. Experiment Design

With the toolkit and basic difficulty measures in place, we conducted a study on the difficulty arising from different Match-3 mechanics and their combinations. Herein, our strategy was built on three decisions: (1) To test mechanics of interest in scenarios reflecting actual game levels rather than trying to isolate them; (2) To aim at scenarios that maximise the information gain about the relationship between mechanics and difficulty; (3) To start with scenarios that capture typical configurations and prevailing design trends. To the latter end, we drew our analysis of level structures in popular titles like "Candy Crush Saga" [14], "Bejeweled" [15], and "Royal Match" [16], as well as from statistical data on gameplay elements in the top 10 grossing Match-3 Games [11].

The resulting test set consists of 20 different scenarios. Aligned with the acronyms introduced in Section III, we label a scenario with collapsing core mechanic, scoring goal and regular deployed pieces with c^{sc}_{reg}. We omit the goal for the majority of scenarios that aims at clearing all regular or some specific pieces. Accordingly, the remaining scenarios are: c_{reg}, c_{rem}, $c_{grw,lck}$, $c_{rem,grw,lck}$, s^{sc}_{reg}, s^{sc}_{mov}, s^{sc}_{blk}, s_{rem}, $s_{blk,rem}$, s_{shd}, $s_{shd,rem}$, $s_{rem,tmb}$, $s_{rem,lck,tmb}$, $s_{rem,lck}$, $s_{grw,lck}$, $s_{grw,lck,rem}$, s_{lck}, $s_{lck,pb}$, and $s_{shd,pb}$. We created three different configurations for each of these scenarios, deploying a small (c_s), a medium (c_m) and a high (c_h) number of special pieces that introduce the respective mechanics. The concrete quantities of special pieces used per configuration were determined based on statistics provided in [11]. To further consistent results, the board size and amount of coloured pieces remained constant across all scenarios. Each configuration was played 10 times by a single tester, resulting in a total number of 30 trials per scenario. Based on the the metrics introduced in Section IV, we calculated difficulty scores (Eqns. 1 and 2) for each scenario's configurations' c_s, c_m and c_h, respectively.

VI. Results

The smallest recorded value emerged in c_s of $c_{rem,grw,lck}$, with a difficulty score of about 0.31. Conversely, the highest difficulty observed emerged in c_h of $\mathscr{s}_{shd}$, receiving a difficulty score of about 2.96.

c^{sc}_{reg} exhibits the lowest difficulty observed, registering a score of about 0.33. Conversely, c_h of $\mathscr{s}^{sc}_{mov}$ reaches the highest difficulty score of about 1.12. In scenarios without special tiles, no different configuration could be distinguished and only one difficulty score was calculated.

The difficulty of all scenarios is plotted in figure 3. We can roughly divide their evolution when scaling the special pieces from c_s over c_m to c_h. Again, the scenarios $\mathscr{s}^{sc}_{reg}$, c_{reg} and c^{sc}_{reg} do not change in difficulty as they do not deploy special pieces. We plotted them nevertheless to visually rank their difficulty in the overall context. In all scenarios that deploy special pieces, the difficulty increases with the first scaling step. We can roughly distinguish between three categories:

- I. The increase in difficulty steepens in the second step ($\mathscr{s}_{shd}$, $\mathscr{s}_{blk,rem}$, $c_{rem,grw,lck}$)
- II. There is an actual drop in difficulty in the second step ($\mathscr{s}_{grw,lck,rem}$, $\mathscr{s}_{rem,lck}$, $\mathscr{s}_{shd,pb}$)
- III. The difficulty still increases but the curve flattens off

Most scenarios fall into case III.

VII. Discussion

In this section, we first attempt to explain the results. Next, we consider the implications for Match-3 game design.

A. Quantitative Analysis

Most scenarios fall into category III, which means that the difficulty increases from c_m to c_h, but the curve flattens. Possibly, the reason for this is that the board gets "saturated" with special pieces at a certain point, and the addition of more pieces creates effects that counteract the added difficulty of more pieces. For example, the addition of more removable obstacles could create a scenario where some of them lie in close proximity to each other. As a result, the player could destroy both with a single match, thus offsetting the difficulty increase of having to deal with more special pieces.

Analogously, in category I-scenarios like $\mathscr{s}_{shd}$, $\mathscr{s}_{blk,rem}$, and $c_{rem,grw,lck}$, with steeper difficulty gain in the second step, single mechanics like shielding might be considered rather hard to begin with and only offset by bomb gadgets and alike, and the constraints introduced by their solitary occurrence just keep getting harder with a rising number of according pieces. Considering $\mathscr{s}_{blk,rem}$, we already see a relatively high difficulty value of $\mathscr{s}_{rem}$ and a modest, fairly constant difficulty in $\mathscr{s}^{sc}_{blk}$. Instead of alleviating their interplay apparently reinforces the challenge—blockers do not make it easier to deal with removable obstacles.

Scenarios belonging to category II exhibit a drop in difficulty from c_m to c_h. This can have multiple reasons. Possibly, in the given, few scenarios, the effects of saturation of the board are strong enough to not only offset the difficulty

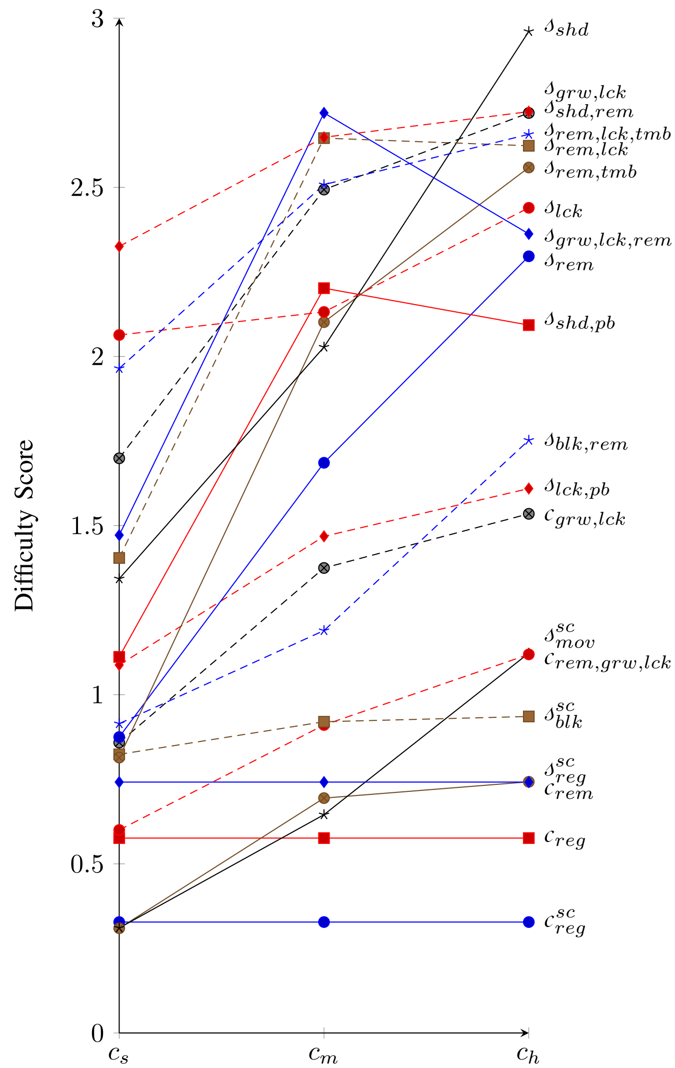


Fig. 3: Difficulty scores of all scenarios plotted. The x-axis represents the different configurations of the scenarios, and the y-axis the magnitude of the difficulty score. The frequency of special pieces used increases along the x-axis. The colors aid with visual clarity and carry no meaning.

increase, but actually lower the difficulty overall. Another possible factor can be outliers in the collected data skewing the results, because of the relatively small sample size.

In cases where the same mechanics are used otherwise, $\mathscr{s}$-scenarios are usually more difficult than c-scenarios. A possible explanation is that collapse is less restrictive than swapping. For collapse, it is sufficient to click a piece connected with an arbitrary amount of equals, even single pieces may be chosen. Hence, the only constraint is the maximization of one's score. A swapping match, however, strictly requires a horizontal or vertical line consisting of at least 3 pieces.

B. Design Ramifications

Based on the results, some conclusions can be drawn for designing Match-3 games. Firstly, based on the difficulty scores of the different configurations of scenarios, it is clear that up to a certain point, a lower amount of special tiles creates an easier scenario. Therefore, the amount of special tiles should be kept low at the beginning of a game. This rather obvious conclusion is also supported by the data found in [11]. Later on, instead of only increasing the number of a single special piece to gradually increase the challenge, a better flow might be established by a concurrent, smooth introduction of other, offsetting pieces. When increasing the difficulty further, the numbers of offsetting pieces might be faded out again, later. The interplay of special pieces needs to be rigorously studied to make the right choices, however, giving the player gadgets to change the board state in more difficult levels, like the paint bomb mechanic, is of course always a safe bet to alleviate the challenge.

In general, the results suggest that the swapping core mechanic requires more balancing than the collapse core mechanic. Score levels also seem to be much lower in difficulty than other scenarios, as much as they can be compared, so using the score goal often early on in a game can be good to familiarize the player with the game play.

Removable obstacles in the $\mathcal{O}$-scenarios seem well-suited for creating more challenging levels beyond the initial stages of the game, without being too frustrating. Shielded pieces, time bombs, and growing obstacles should be kept to the later stages of the game. Lastly, locked elements should only be used in small quantities or be kept to the most challenging levels of the game, especially when using the swapping core mechanic.

VIII. Summary and Future Work

In this paper, an open-source toolkit was introduced that allows for easy creation of many Match-3 game scenarios, and subsequent testing and data collection. This toolkit was used to conduct a study on the effects of different Match-3 mechanics on the difficulty of a scenario. The results showed that the swapping core mechanic creates more challenging game scenarios than the collapse core mechanic. Allowing the player to alter the board state, for example by changing the color of certain tiles, has the rather obvious effect of reducing the difficulty also in otherwise quite challenging scenarios. There are, however, also special pieces, or mechanics, whose interplay's effect on the difficulty should be studied more rigorously to ensure a well-balanced game flow.

As future work, other, more refined difficulty functions could be developed, e.g. that can evaluate both score and non-score scenarios to allow for a better comparison between these scenarios. To this end, a machine learning agent to automatically evaluate the difficulty and level design of scenarios as proposed by [17] or [18] could also be employed. Such automated testing would also give more leeway for more thorough testing, e.g. with more scenarios, continuous changes in the deployed pieces and, as a result, clearer understanding of the tipping points of difficulty evolution. Complementary, more user-based studies could help to validate the proposed difficulty functions to begin with, and to gain more accurate insights on more mechanics and their combinations. Especially the inclusion of more core mechanics could be interesting, and to see whether the learnings about the design of Match-3 games reported in this paper hold true for games with other control types.

References

[1] D. Curry, "Candy crush revenue and usage statistics (2023)," https://www.businessofapps.com/data/candy-crush-statistics/, 2023, accessed: 2023-12-18.

[2] J. Clement, "Most popular match-3 gaming apps worldwide in july 2023, by downloads," https://www.statista.com/statistics/1405591/most-downloaded-match-3-games-worldwide/, 2023, accessed: 2024-01-10.

[3] ——, "Most popular match-3 gaming apps worldwide in july 2023, by revenue," https://www.statista.com/statistics/1405596/top-grossing-match-3-gaming-titles/, 2023, accessed: 2024-01-10.

[4] E. Kiiski, "Match3 – meta layers and matching types," https://www.gamerefinery.com/match3-meta-layers-matching-types/, 2020, accessed: 2023-05-31.

[5] K. Razumovskiy, "45 match-3 mechanics," https://www.gamedeveloper.com/design/45-match-3-mechanics, 2018, accessed: 2023-05-31.

[6] AppLovin, "What are match-three games?" https://www.applovin.com/blog/what-are-match-3-games/, 2020, accessed: 2023-12-18.

[7] Starloop Studios, "Match-3 games: What, how and why," https://starloopstudios.com/match-3-games-what-how-and-why/, accessed: 2023-05-31.

[8] J. Fraser, M. Katchabaw, and R. E. Mercer, "A methodological approach to identifying and quantifying video game difficulty factors," *Entertainment Computing*, vol. 5, no. 4, pp. 441–449, Dec. 2014.

[9] M.-V. Aponte, G. Levieux, and S. Natkin, "Measuring the level of difficulty in single player video games," *Entertainment Computing*, vol. 2, no. 4, pp. 205–213, Jan. 2011.

[10] E. M. Varonis and M. E. Varonis, "Deconstructing Candy Crush: what instructional design can learn from game design," in *The International Journal of Information and Learning Technology*, D. Gregory S. Anderson, Ed., vol. 32, Jun. 2015, pp. 150–164, iSSN: 2056-4880 Issue: 3.

[11] A. Giokarini, "Level design complexity in match-three games: a study of level design structural complexity in modern, casual, mobile match-three puzzle games," Master's thesis, Taiteiden ja suunnittelun korkeakoulu, 2020.

[12] Godot Foundation, "Godot engine." [Online]. Available: https://godotengine.org/

[13] T. Pavlidis, "Algorithms for Graphics and Image Processing." Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. [Online]. Available: http://link.springer.com/10.1007/978-3-642-93208-3

[14] King, "Candy crush saga," [Android-App], 2012.

[15] PopCap Games, "Bejeweled," [Browser-App], 2001.

[16] Dream Games, "Royal match," [Android-App], 2020.

[17] Y. Shin, J. Kim, K. Jin, and Y. B. Kim, "Playtesting in Match 3 Game Using Strategic Plays via Reinforcement Learning," *IEEE Access*, vol. 8, pp. 51 593–51 600, 2020.

[18] S. A. Dukkancı, "Level generation using genetic algorithms and difficulty testing using reinforcement learning in match-3 game," Master's thesis, Middle East Technical University, Sep. 2021.