Tampere University of Applied Sciences

Klondike Solitaire Solvability

Mikko Voima

BACHELOR’S THESIS
April 2021

Degree Programme in Business Information Systems
Option of Game Development

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Option of Game Development

VOIMA, MIKKO:
Klondike Solitaire Solvability

Bachelor's thesis 32 pages, of which appendices 1 page
June 2021

Klondike solitaire remains one of the most popular single-player card games, but the exact odds of winning were discovered as late as 2019. The objective of this thesis was to study Klondike solitaire solvability from the game design point of view. The purpose of this thesis was to develop a solitaire prototype and use it as a testbed to study the solvability of Klondike.

The theoretical section explores the card game literature and the academic studies on the solvability of Klondike solitaire. Furthermore, Klondike solitaire rule variations and the game mechanics are analysed. In the practical section a Klondike game prototype was developed using Unity game engine. A new fast recursive method was developed which can detect 2.24% of random card configurations as unsolvable without simulating any moves.

The study indicates that determining the solvability of Klondike is a computationally complex NP-complete problem. Earlier studies proved empirically that approximately 82% of the card configurations are solvable. The method developed in this thesis could detect over 12% of the unsolvable card configurations without making any moves. The method can be used to narrow the search space of brute-force searches and applied to other problems. Analytical research on Klondike solvability is called for because the optimal strategy is still not known.

Keywords: card games, computational complexity, game design

CONTENTS

1 INTRODUCTION

Klondike is probably the most played single player card game in the world. Despite this, the mechanics that make a game of Klondike solvable or unsolvable remain little-studied.

The main goal of this thesis is to study the solvability of Klondike from a game design point of view and analyse the underlying game mechanics. The thesis seeks answers to questions such as: “What makes an initial card configuration solvable or unsolvable?”, “What choices a player makes during a game of Klondike?”, and “How do the player’s choices lead the game into a winnable or unwinnable state?”

In chapter 2, a brief history of solitaire games and Klondike solitaire is given. How the game was popularised by Microsoft solitaire and possible reasons for the popularity are also considered.

Chapter 3 summarises the studies in the academic literature on Klondike solvability and computational complexity. A short explanation of the computation complexity of Klondike is given as it relates to the difficulty of the problem. The methods used in previous studies and the results are referenced.

Then the Klondike rules are clearly defined in Chapter 4, and different rule variants are addressed. The analysis of the game mechanics starts by defining the solvability of an individual card. The player choices and how they affect the outcome of the game are addressed next.

Chapter 5 details the development of a playable Klondike solitaire prototype using Unity game engine. It is used as a testbed to develop a recursive method for tracing the card solution requirements. The recursive method is then used to analyse a large sample of initial card configurations to detect unsolvable games. Lastly the conclusions are drawn and discussed.

2 HISTORY

Klondike solitaire has a long history of continued popularity even before Microsoft Solitaire. Understanding the possible gambling roots is essential to understanding Klondike, as it puts the questions of naming, solvability, and rule variants into the right frame of reference. The introduction of Microsoft Solitaire then popularised the game even more in the 1990's.

2.1 History of solitaire card games

The exact origins of single player card games are uncertain. Games scholar and historian David Parlett (n.d.) asserts German or Scandinavian origin as the most probable, citing multiple other authors. The oldest known book of solitaire games was published in Moscow in 1826, and at least six more were published in Sweden before the year 1850. Over the course of the 19^{th} century more and more books about solitaire card games were written and translated into different languages thus spreading solitaire card games all over the world. The first known solitaire book written in English was Lady Cadogan's Illustrated Games of Solitaire or Patience in 1870. (Parlett n.d..)

A likely predecessor to Klondike solitaire called Triangle appears at least as far back as 1895 in Ednah Cheney's Patience: a series of games with cards (Cheney 1895, 117–119). Some years later a more familiar version called Canfield or Klondike appears in later revisions of Lady Cadogan's Illustrated games of Solitaire or Patience (see Figure 1) (Cadogan 1914, 118). The game rules present counters as bets like in a gambling game. The player would pay 52 counters for the pack of cards and would win 5 counters for every card played into the foundation (Cadogan 1914, 118–119). The name Klondike likely refers to the Klondike gold rush likening the chances of winning to that of finding gold.

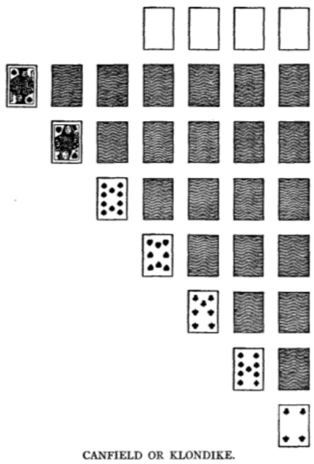


FIGURE 1. Illustrated Canfield or Klondike (Cadogan 1914, 118).

There is confusion surrounding the name Canfield and gambling origins. Richard Canfield ran illegal gambling casinos throughout North-eastern United States in the late 19th and early 20th century. During that period a solitaire card game was played as a casino game. Card game disseminator and writer Richard Foster (1914, 512) claims that Klondike and Canfield are two different card games often confused with one another. Betting in Canfield with dollars is explained implying this to be the casino game of the two (Foster 1914, 513, 694). The game Foster (1914, 696) calls Canfield is known as Fascination (Dick 1896, 10) or Demon Patience (Hapgood 1920, 37–39) in the other sources from the period. In modern times the game is known as Canfield in America and as Demon in England (Parlett 1980, 94).

Bridge players and writers Morehead and Mott-Smith wrote a book called The Complete Book of Solitaire and Patience Games in 1949. In it they explicitly state the game played at Canfield's casino was Canfield and not Klondike citing Foster as a source (Morehead & Mott-Smith 2015). However, the author of the Solitaire Laboratory website Michael Keller makes a compelling argument suggesting it may have been Klondike that was played at Canfield's casinos citing numerous solitaire authors from early 1900's (Keller 2013). If indeed it is Foster who is mistaken, and Demon was not played at Canfield's casino, that would certainly explain why Lady Cadogan (1914, 118) and Hapgood (1920, 180–182) both refer to Klondike as Canfield and why both feature gambling inspired scoring or counters.

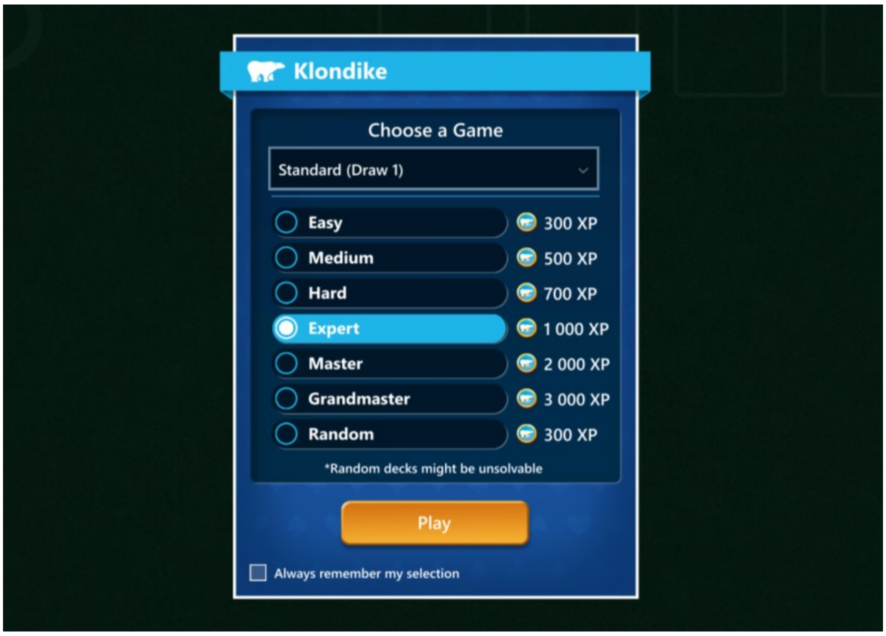
According to Morehead and Mott-Smith (2015), by the 1950’s Klondike had already become the most played solitaire card game by far. Klondike was still widely known as Canfield at the time, or simply called *solitaire* either for being the only known solitaire or because it was mistaken to be the proper name for the game of Klondike itself. Morehead and Mott-Smith are perplexed by Klondike’s popularity calling it an inexplicable mystery. (Morehead & Mott-Smith 2015.) “Why the game itself should have achieved such popularity defies explanation” (Parlett 1980, 94).

2.2 Microsoft Solitaire

The digital version of Klondike solitaire: Microsoft Solitaire was developed by a Microsoft intern Wes Cherry in the summer of 1988. The game was subsequently bundled into Windows 3.0 operating system in 1990 and gained considerable popularity. Officially it was intended to help users to learn to use the computer mouse, which was novel at the time. (Meet the Intern Who Wrote Solitaire for Microsoft 2017.) In the process, Microsoft Solitaire introduced digital games to the large and diverse gaming audience now known as the casual gaming market.

Microsoft Solitaire is often referred as an example of a casual game. Game researcher Annakaisa Kultima (2009) proposes that the rise of casual gaming should more accurately be characterised as “the normalization of digital play”. The wide appeal of solitaire could be attributed to the low barrier of entry and the familiarity with physical card games. (Kultima 2009, 58–63.) Solitaire came pre-installed on Windows operating systems from Windows 3.0 in 1990 until Windows 7.

Developed by Smoking Guns Interactive the Microsoft Solitaire Collection was introduced with Windows 8 in 2012 as a separate download on the Microsoft Store. It bundled previous Microsoft Solitaire, FreeCell, and Spider solitaires together and added Pyramid and TriPeaks solitaires. More new features such as daily challenges were introduced in subsequent updates. Guaranteed solvable decks were added in an update in late 2016 to allow players to choose a suitable difficulty level (Microsoft Solitaire Collection: Solvable Decks! 2017). Picture 1 shows the screen where the difficulty level of the solvable deck is chosen. The game was also ported outside Windows to Android and iOS in 2016.



PICTURE 1. Screenshot from Microsoft Solitaire Collection where the difficulty of the solvable deck is chosen.

As of May 2020, Microsoft Solitaire hosted 35 million players each month (Jensen 2020). Likewise, in the Finnish study *Pelaajabarometri*, solitaire maintained its position as one of the most popular digital games in Finland (Kinnunen, Taskinen & Mäyrä 2020). In 2019 Microsoft Solitaire was officially inducted into the Video Game Hall of Fame for its influence, longevity, geographical reach, and icon-status (The Strong National Museum of Play 2019). Despite being often overlooked as commonplace, solitaire continues to appeal to a large and diverse gaming audience.

3 SOLVABILITY

Finding a solution to Klondike is a complex problem that seems to require attempting large amounts of combinations of choices. Because of that, card game books usually provide only vague estimates of the chances of winning solitaire games. Solving Klondike with computer algorithms has been attempted many times with varying success. A study in 2004 could finally win twice as many Klondike games as an expert human player. Subsequent studies have reached even better results with increases in computer performance and newer programming techniques.

3.1 History

Historically solitaire card games may be related to fortune-telling with tarot cards as suggested by Parlett (n.d.). As such it was up to the player to decide if the contest was against fate, chance, or circumstance (Cheney 1895, 6). However, it was understood that some games required more skill or ingenuity than others. Cheney (1895) writes of Triangle "Much skill can be employed in this game, and it may often be won by careful changes and rearrangements when it seems to be lost." Chances of winning are described subjectively and vaguely: "succeeds often enough to encourage the player". (Cheney 1895, 117–119.)

With much stricter rules of Klondike, the player was estimated to win "one in 30 games" (Morehead & Mott-Smith 2015). Parlett (1980, 11–16) further analyses why some solitaire games are harder to win than others and how much skill could be employed. The analysis identifies the distribution of cards and the strictness of the game rules as major factors affecting winnability. Strict rules limit the choices available whereas lax rules make the card manipulation too easy. A suggested ideal game should be winnable regardless of the card distribution. However, this is deemed unprovable without an extensive computer analysis. (Parlett 1980, 19–23.)

In 1999 mathematicians David Aldous and Persi Diaconis (1999, 1–25) examined a simple solitaire card game called *patience sorting* for which the optimal strategy is known. The paper found a surprising connection between patience sorting and random matrices which were initially introduced to model the nuclei of heavy atoms. In the final remarks the authors find “surprisingly little serious effort to analyze other solitaire games” and list ongoing research at the time of writing. (Aldous & Diaconis 1999, 1–25.) In these early studies computer algorithms could not beat expert human players.

3.2 Computational complexity

In computer science, computational complexity describes how the amount of resources required to solve a given problem scale with the size of the problem. One of the most important resources is the computation time. Problems that can be solved in polynomial amount of computation time, or polynomial time, are generally quick to solve. In comparison, problems that require exponential amount of computation time, or exponential time, are generally more difficult or time-consuming problems to solve.

In a paper called The Complexity of Solitaire Luc Longpré and Pierre McKenzie (2009) investigated the computational complexity of Klondike solitaire. They specifically examined the complexity of determining, whether an initial game configuration of an n-card Klondike can be won. The problem was shown to be NP-complete. (Longpré & McKenzie 2009, 1–8.)

NP stands for Nondeterministic Polynomial time and can be solved on a Nondeterministic Turing Machine (NTM) in polynomial time. Unfortunately, NTM is a theoretical hypothetical model of computing that is impossible on any currently known real computer. Demaine (2013) describes NTM as a “lucky” algorithm that always guesses the correct choice if a solution exists. (Demaine 2013.) This is equivalent to winning all the solvable Klondike games on the first try.

Real computers can check if a set of moves will solve a game of Klondike in polynomial time. However, finding a solution with a deterministic algorithm is much harder. At worst, finding a solution could require trying out all possible combinations of moves which is called a brute-force search. A brute-force search takes exponential time but is guaranteed to find a solution if one exists. NP-complete problems are commonly thought to unsolvable in polynomial time, but computer scientists have not been able to prove it (Demaine 2013).

> On the other hand, Klondike being NP-complete provides absolutely no mathematical justification that investigating the odds of winning in the case of a standard 52-card deck will be difficult. But the fact that Klondike is just another name for SAT can at least be seen as confirmation that the game does involve a good level of intricacy. (Longpré & McKenzie 2009, 8)

A standard 52-card Klondike is ultimately a finite problem given the finite number of cards, and card stacks, and hence, the number of possible moves is also finite. However, the number of all possible card configurations is still the factorial of 52 or about 8×10^{67}. Likewise, the number of different combinations of possible moves within each game is so large that brute forcing through all possible combinations is not practical. However, several studies, aided by the advances in computer performance and dynamic programming techniques, have been conducted on Klondike win rate.

3.3 Applied mathematics

A 2004 paper entitled *Solitaire: Man Versus Machine* was the first to use a rollout method to analyse Klondike solitaire. The paper calls the inability to determine the odds of winning Klondike “one of the embarrassments of applied mathematics”. In the study a senior mathematician had played 2,000 games and carefully recorded the results achieving a win rate of 36.6%. This study chose to study a Klondike variant where all cards are known from the start effectively coining the term *thoughtful Klondike*. (Yan, Diaconis, Rusmevichientong & Van Roy 2004, 1–7.)

In the standard Klondike the locations of all the cards are not known at the start. This forces the player make blind choices. In thoughtful Klondike the only difference is that all the cards are known right at the start. Any solution to thoughtful Klondike also works for standard Klondike and vice versa. The thoughtful variant allows for deductions to be made and is therefore considered more thought-provoking and challenging (Yan et al. 2004, 7).

A heuristic strategy used to score legal moves was based on Microsoft Solitaire scoring with further adjustments based on intuitive understanding of the game mechanics. Initially the algorithm could only win 13.1% of the games on the first try. The performance was then improved with a rollout method whereby at the start of the game the entire remainder of the game would be simulated with the heuristic strategy for each of the legal moves available. This improved the win rate to 31.2%. (Yan et al. 2004, 1–7.)

The performance was further improved by adding more rollout iterations. However, the computation time increases exponentially by the number of iterations. At 5 rollout iterations each game would take 105 minutes on average and only 200 games could be played due to time constraints achieving a win rate of 70.2%. (Yan et al. 2004, 1–7.) These results indicate that the early moves in a game affect the win rate considerably. Increasing the number of rollouts further would ultimately iterate through all possible combinations of legal moves and thus ultimately find an optimal strategy.

Yan et al. (2004) also attempted to figure out what percentage of the deals could possibly be solvable by enumerating unsolvable card configurations. They were able to rigorously prove that at most 98.81% of card configurations are solvable or conversely that 1.19% could be proven unsolvable (Yan et al. 2004, 1–7). These percentages suggest it is easier to prove a deal solvable than it is to prove that a deal is unsolvable. A single solution proves a deal solvable, whereas proving a deal to be impossible analytically is hard.

Next study to improve the win rate for thoughtful Klondike was Searching Solitaire in Real Time by Bjarnason, Tadepalli, and Fern (2007a). It implemented multistage nested rollouts so that computational resources could be better allocated. “By tuning the level of rollout search for each stage, resources can be allocated to difficult stages and conserved in simple stages, with an appropriate heuristic applied to each stage." By using these techniques, the authors were able to win 74.9% of games in less than 4 seconds of computation time per game. (Bjarnason, Tadepalli & Fern 2007a, 1–12.)

Bjarnason et al. (2007a) also experimented with more degrees of nesting. The most successful configuration won 82.24% of time deals in an average time of 31.67 minutes per game. The average time per win was 0.44 minutes and 176.27 minutes per loss. (Bjarnason, Tadepalli & Fern 2007a, 1–12.) With wins, the search could be stopped as soon as a solution was found. The lost games are the more time-consuming part due to the exhaustive search.

The same authors (2007b) wrote another paper this time aiming to improve win rates for the standard probabilistic Klondike rather than the deterministic thoughtful Klondike variant. This time they utilised probabilistic planning techniques such as Hindsight Optimisation (HOP) and Upper Confidence bounds applied to Trees (UCT) which applies a particular bandit algorithm for rollout-based Monte-Carlo planning. These approaches can win up to 35% of random games with little prior knowledge. "A better theoretical understanding of why these algorithms are so successful would be very valuable." (Bjarnason, Tadepalli & Fern 2007b, 1–8.)

The latest and most complete effort to find out the chances of winning Klondike solitaire comes from at the time undergraduate student Charlie Blake and the professor of computer science at St. Andrews University Ian Gent (2019). They implemented a general-purpose AI solver for 48 different solitaire card games with configurable rulesets. The solver is based on depth-first backtracking search and utilises several AI techniques to improve efficiency. Priority was given to obtaining definite answers about whether a given card configuration is winnable or not. (Blake & Gent 2019, 1–11.)

Large scale experiments were conducted on the Edinburgh Parallel Computing Centre (EPCC) supercomputers using approximately 20 years of CPU-time in total and 2.4 years of CPU-time for Klondike. Of the one million games of thoughtful Klondike analysed 81,936% were wins with the solver averaging 29 million nodes per instance of the game. In the experimental results 18.024% were losses, leaving only a very small percentage of games as unknown due to timeouts. The final conservative estimate was placed in the middle between these figures at 81.956%±0.096%. Blake's and Gent's (2019) paper *Winnability of Klondike Solitaire* contains the full results for all games analysed and the techniques used. (Blake & Gent 2019, 1–11.)

> While greatly improving the state of knowledge, many important questions remain open. Perhaps the two most fundamental are: first, to give analytic results instead of the purely empirical ones we have given; and second, to calculate the winnability of games like Klondike and Canfield in their non-thoughtful variants. Both of these questions are so little-studied in the literature that it is hard to even estimate how difficult they are. (Blake & Gent 2019, 7.)

In conclusion brute-force analysis of the game of Klondike that exhausts all the possible combinations of moves to prove a game winnable or unwinnable remains computationally expensive option. The more advanced AI techniques greatly increase the efficiency by identifying equivalent game states and redundant moves. However, they do not explain what makes a game winnable but rather which ones are winnable. Thus, an analysis of the game mechanics of Klondike is warranted.

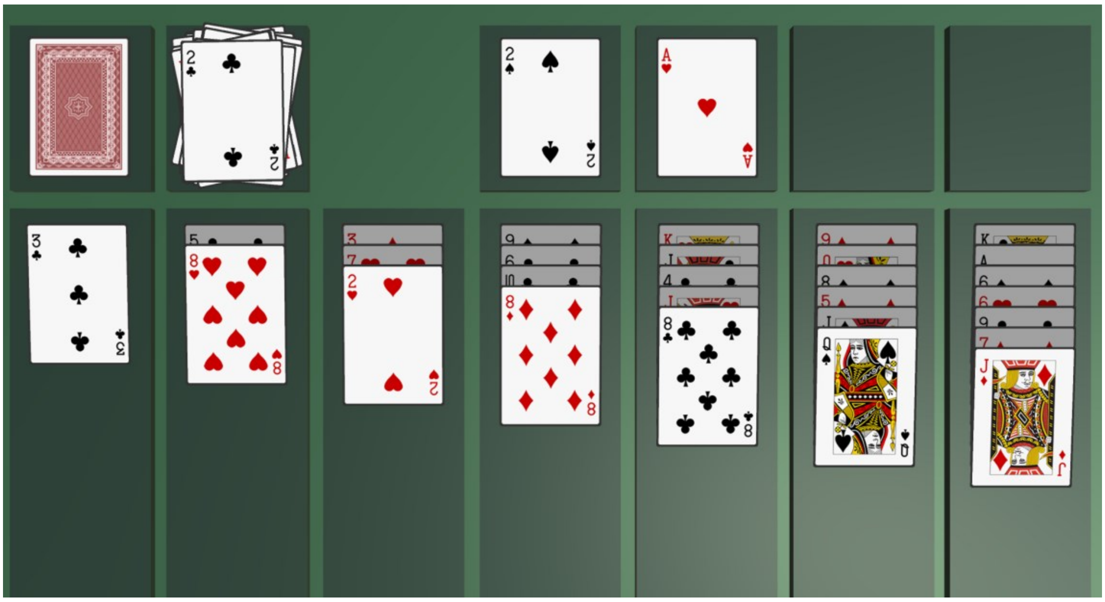
4 RULES OF KLONDIKE SOLITAIRE

The exact game rules followed affect the solvability and perceived difficulty of any given game. In physical card games the players can choose to adopt different rules independently and these are often called house rules. Card game books typically also mention alternative variants to make a game easier or harder. How exactly different rule implementations affect solvability percentages is not known. For a game prototype version of Klondike solitaire, the precise rules need to be clearly defined. Different rule variants are considered, and the variant that allows for the most options or player choices is chosen.

4.1 Game rules and rule variants

Klondike solitaire is played with a single standard 52-card deck with clubs and spades as the black suits, and hearts and diamonds as the red suits. Each suit consists of 13 cards ranked 1 through 13. Ace ranks as one, whereas Jack, Queen, and King rank as 11, 12, and 13 respectively. In Klondike there are thirteen piles in total: seven build piles, four foundation piles, a stock, and a waste pile.

The thoughtful variant is selected for the game prototype as it allows for deductions to be made. In the thoughtful variant all cards are laid face up. The cards that would be laid face down in the non-thoughtful variant are called closed as they are unavailable for play. Opening a card refers to a card becoming available or open and corresponds to turning a card face up in the non-thoughtful variant. At the beginning of the game, seven build piles are laid horizontally on the table left to right and fanned down towards the player. One card is laid on the first pile, two are laid on the second, and so on until seven cards are laid on the seventh pile. The top card of each build pile is open and all the cards under it are closed. Picture 2 shows the open cards as white and the closed cards as greyed out. Build piles are numbered one through seven from left to right.



PICTURE 2. Card layout for thoughtful Klondike.

Cards can be packed on the build piles if two conditions are met. The card placed on top must be one lower in rank and of the opposite colour: black on red and red on black. The cards below are blocked by the new top card. If any closed cards are revealed, they are opened. Sequences of cards can be packed on the other build piles if the top card of the receiving build pile is of the opposite colour and one rank higher than the bottom card of the sequence of cards being packed. The older rules do not allow for the sequences of packed cards to be split: “all must be moved together or not at all” (Cadogan 1914, 119).

Over time the rules about packing sequences became more lenient. By the 1950’s some players had started to allow the top card to be packed even if it had other packed cards under it (Morehead & Mott-Smith 2015). Decades later, splitting sequences of packed cards or packing partial sequences had become the norm and instead some players would disallow it (Parlett 1980, 95). In modern times all versions of the Microsoft Solitaire allow packing partial card sequences between build piles. This allows for the most opportunities to manipulate the cards and was the rule interpretation of choice.

The goal is to organise all cards into four foundation piles in ascending rank following suit. The four foundation piles are laid side by side above the row of build piles. Each foundation pile starts with an Ace as they are encountered during the game. Only cards one rank higher of the same suit can be founded on top. The top cards from the waste pile and the seven build piles are eligible to be founded into the foundation piles. The top card of each foundation pile can be moved to the build piles adhering to the packing rules.

Whenever all the open cards of any build pile are founded or packed to the other build piles and a closed card is revealed, that closed card is opened and becomes usable. If no cards are left in a build pile it is known as a space and can be filled with a King or a sequence of cards that begins with a King. As opening closed cards opens new possibilities for moves it should be considered the primary objective in the game. This is how all the 52 cards are made available so that they can be founded into the foundation piles.

At the start of the game 28 cards are dealt (as described) into the build piles and the leftover 24 cards form the stock. In the draw-1 variants the stock is gone through one card at a time. Draw-3 forms another family of variants where cards are drawn three at a time. Additionally, in some variants the number of times the stock can be gone through is limited and unlimited in others. In each variant however, the stock cards are moved to the top of the waste pile. The top card of the waste pile is always usable.

In the gambling variant the stock can be run through only once, one card at a time (Cadogan 1914, 119). In this variant the player pays 52 counters to play the game and wins 5 for each card played into the foundation (Cadogan 1914, 119; Keller 2013). With these rules the chances of winning are deliberately low so that the house would run a profit. The chances of winning in this variant are estimated at 1 in 30 games (Morehead & Mott-Smith 2015). In Microsoft Solitaire this variant is known as the Vegas game mode (Keller 2013).

In the draw-3 variant the stock is drawn three cards at a time, and only the topmost of the three cards is usable. The covered cards become usable as they are revealed. In this variant the stock can be run through three times (Parlett 1980, 95; Morehead & Mott-Smith 2015) or in other version unlimited number of times (Foster 1914, 512; Hapgood 1920, 181). In this variant, card availability is dependent on the order in which the cards appear which may prevent solvability in some cases. The academic literature has studied the solvability of the 3-draw variant with unlimited runs of the stock.

In the most lenient variant, the stock is drawn one card at a time and can be run through unlimited number of times. This variant was likely popularised by Microsoft solitaire and does not appear in the major card game books. Under these rules all cards in the stock are always available which affords the best chances of winning. In this thesis the draw-1 with unlimited stock runs is analysed with focus on the initial card configuration solvability and the player choices during gameplay.

4.2 Single card solvability

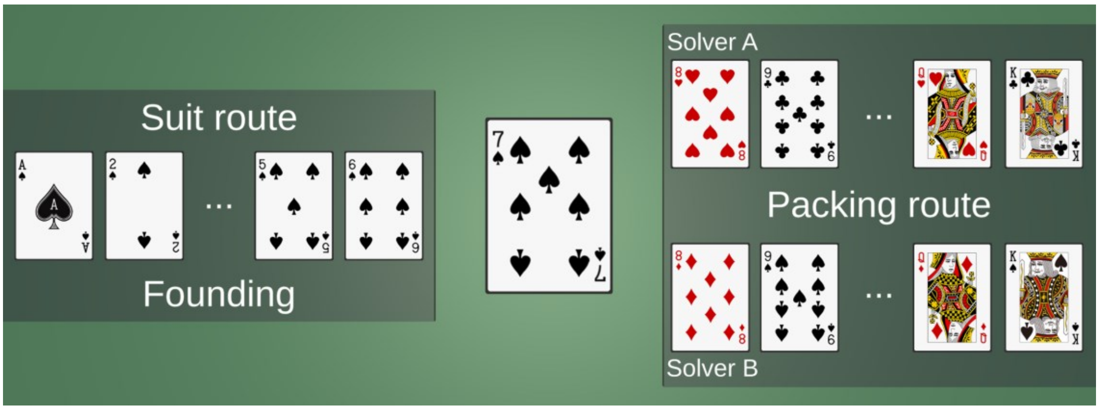
A game of Klondike can be won only if all the closed cards can be opened. In that sense opening each of the closed cards is equally important. A closed card can be opened if the card directly above it can be solved by founding or by packing. Depending on the position in a build pile a card can block some or all of these solve routes. A card that blocks its foundation solution route by blocking a lower rank card of the same suit is called suit blocked. An Ace cannot be suit blocked and is always solvable whereas a King requires an empty space to be solved.

For a packing solution any card except a King has two direct solvers of the opposite colour and one rank higher called solver A and solver B. These too can be blocked under the card itself thus limiting solution options. If a card is suit blocked and both direct solvers are blocked, then the card is impossible to solve making the whole game unsolvable. The probabilities of a card being free, suit blocked, solver blocked, or both were calculated for a single pile of up to seven cards. All the possible permutations were considered and assumed to be equally likely. The results are presented in Table 1.

TABLE 1. Probabilities of different types of solve blocks per card.

Pile height	2	3	4	5	6	7
No solver blocked	84.92%	72.42%	62.06%	53.47%	46.33%	40.39%
Suit blocked	11.76%	21.01%	28.18%	33.65%	37.74%	40.70%
One solver blocked	3.32%	5.71%	7.38%	8.50%	9.20%	9.59%
Solver & suit	–	0.80%	2.18%	3.98%	6.06%	8.33%
Both solvers	–	0.07%	0.17%	0.31%	0.45%	0.60%
Both solvers & suit	–	–	0.02%	0.09%	0.21%	0.40%

A card is overwhelmingly more likely to be suit blocked than solver blocked. However, these percentages are from a single card and a single pile perspective and only offer a simplistic view. These blocks describe which solution routes are known to be unavailable, not which solution route is the right one or even available on the other build piles. A suit solution, solver A, and solver B should be thought of as alternative solution routes one of which must be satisfied. If a solution is not directly available, a card requires another card or multiple cards to be solved first. These solution prerequisites are called card requirements. Each solution route available has different requirements (Picture 3).



PICTURE 3. Solution routes visualised.

4.3 Player choices

For optimal play all moves should be goal oriented for the purpose of opening a closed card by solving the card above. By choosing which of the currently solvable cards is solved, the player implicitly chooses which of the closed cards is opened. If there are multiple ways to solve a card, the player also chooses whether the solvable card is packed or founded. Packing may involve even further choices of which suits are used for packing.

In a successful game, earlier choices fulfil the card solution requirements for subsequent moves until all the closed cards are opened. At the end of the game, a card requirement chain could be traced back to the start. In contrast in an unsuccessful game, earlier moves violate the card requirements in such a way as to make one or several cards impossible to solve.

Figure 2 visualises the choices made in the Example Game 1 (Appendix 1). In this example game, the choices of which suits are used in packing affect the outcome. For a human player these choices are easy to miss if the player does not realise the stock often offers two suits as options. At a crucial moment either Jack of Hearts or Jack of Diamonds must be founded for the game to come out. Depending on which suits were used in packing earlier on, both red Jacks can become impossible to found.

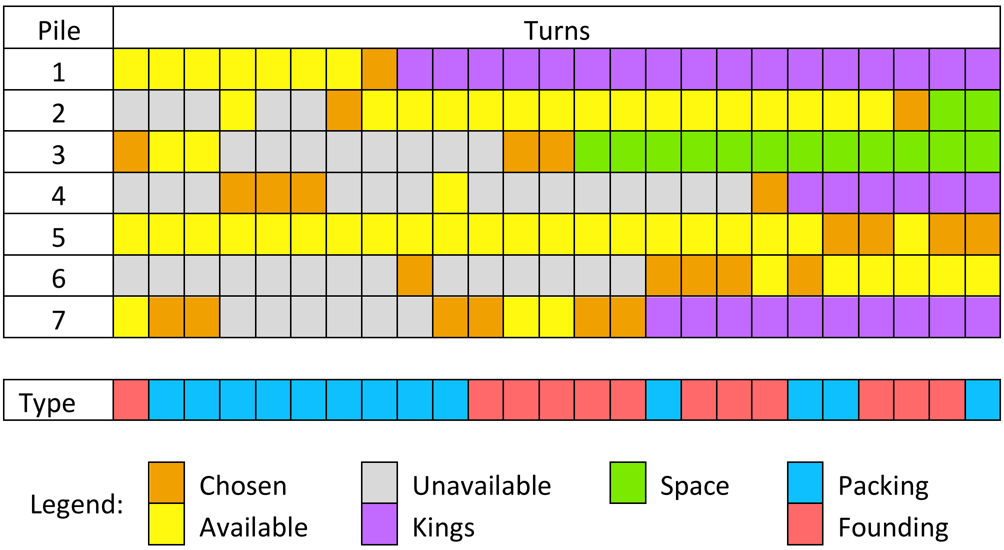


FIGURE 2. Decision diagram for example game 1.

Predicting which choices, if any, lead to a solution is a difficult problem. The earlier choices in the game can both block future choices and open new opportunities. The time-consuming way is to try different combinations of choices until the game is either solved or all possibilities are exhausted. Some games can also be proven to be unsolvable faster by identifying self-contradicting solution requirement such as a card requiring itself to be solved first to solve.

5 KLONDIKE PROTOTYPE IMPLEMENTATION

A playable Klondike solitaire prototype game was developed to study the game in practice. The prototype was implemented in Unity game engine due to easy prototyping and author's familiarity with the engine. The prototype also served as a testbed for card solution algorithm development to visualise the problem.

5.1 Design and development

Unity long term support version 2019.4.23f1 was chosen for this project. A ready-made playing cards asset was purchased from Unity Asset Store to provide the card graphics. The asset used proved well optimised as all the cards shared the texture. In practice this means the draw calls could be batched together to decrease the time spent on rendering. An unlit shader was used for the cards and shadow casting was disabled to speed up the rendering even more.

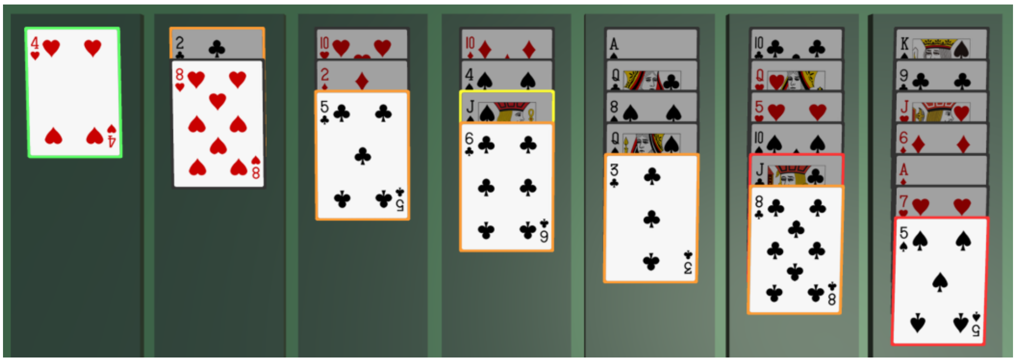
Standard Unity features were utilised to develop a working prototype quickly. For example, the card pile visuals are simply scaled boxes with a different colour. These boxes also doubled as the input targets and were scaled up to make them easier to hit. The game is designed to be mouse controlled with added features for convenience. The stock can be run through forwards and backwards with the mouse wheel so that different approaches can be tried quickly.

The game logic was scripted using the C# programming language with a modular design. A central Game Master class initialises all the other modules and manages the high-level functions like generating new card configurations, saving, loading, and tracking game progress. Smaller modules such as player functions, statistics or move history are initialised with a static instance for easy access in the other classes.

A card pile class implements the basic functions of receiving, storing, and sending cards. Each individual pile type inherits these basic features and add their own pile type-specific features with overrides. The player class checks the source and target piles of the player moves by casting rays to see which piles were hit. The move validity is verified from the target pile by asking if a particular card is accepted. To shuffle the cards in the stock, the Fisher–Yates shuffle algorithm was implemented. Unity standard random library was used, which uses a Xorshift 128 algorithm to generate pseudorandom numbers. It is initialised with a high entropy seed from the operating system (Unity - Scripting API: Random 2021).

The individual card class became the central focus as it could store references to its direct solvers, direct neighbours, and suit cards both up and down in rank. Card blocks are determined here by the cards themselves. Various messages and checks can be sent recursively from a starting card to the relevant cards in the chain.

To visualise the various card states and conditions an outline was added that could be coloured via shader parameters. Colouring the card itself allowed even more features to be visualised. In Picture 4 the green outline means the card can be directly founded. The eight of Clubs is detected as only solvable to foundation and the condition is visualised with an orange outline. The outline is meant as a warning to avoid blocking the Clubs to keep the eight of Clubs solvable and the warning is passed down in rank recursively. The red outline denotes that a card blocks one of its direct solvers. For example, the black Jack in Pile 6 should be solved first. A yellow outline denotes a lower priority implying the black Jack in pile 4 should be solved second.



PICTURE 4. Various card highlight effects active.

A recursive method to trace the card solution requirements was developed. The simplified operating principle is depicted in Figure 3. The solvability of card X is examined, and it requires either card A or B to solve. Card A is blocked under card X, but card B can be used. For card B to solve card X, the card B needs to be opened first and hence all the cards above B need to be solved out of the way. The card directly above card B is card F which needs either card G or H to solve. Both card H and card G are blocked by cards F and X respectively and F cannot be solved before X. Card X cannot be solved before F and therefore X is unsolvable.

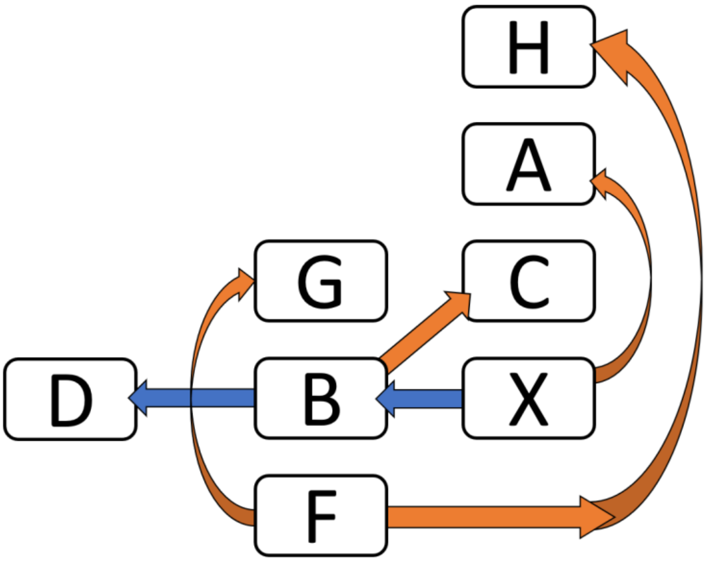
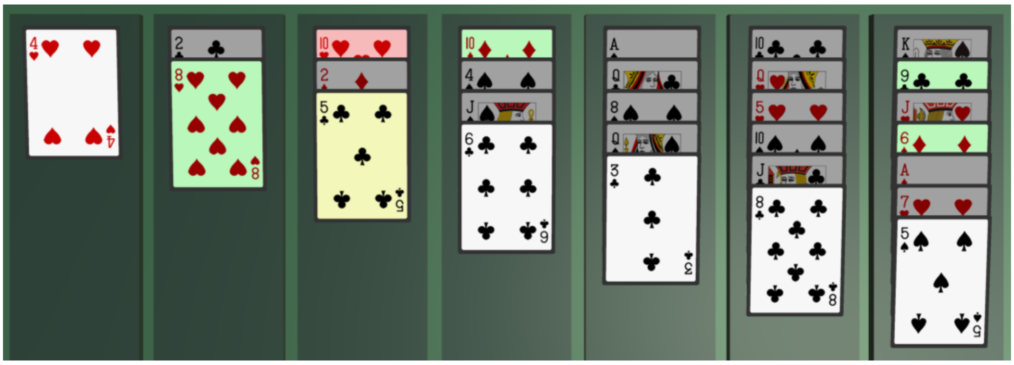


FIGURE 3. The recursive method operating diagram.

If one or both solvers are on the stock, the solvers of solvers can be traced recursively until a card onto which the solver can be ultimately packed is reached. At this level all options are exhausted. However, if card H were to be in a pile other than F pile or X pile, it would simply be assumed to be reachable to limit the search depth. If the packing solutions are exhausted a Foundation solution is traced instead. Each card down in suit needs to be reachable all the way down to Ace without solving card X first.

The recursive method can also be used in game to look for non-trivial card solution routes by pressing the middle mouse button or mouse wheel while hovering over a card. In Picture 5, the card being traced is coloured yellow, blocked solvers in red, and possible solvers in green. This feature proved useful during development to verify the recursive method was working correctly. If only the suit solution route was left, the suit cards would be highlighted instead.



PICTURE 5. Possible solvers of Five of Clubs highlighted.

Winning the game is celebrated with a new take on the classic Microsoft solitaire victory animation (Picture 6). The card trail effect is achieved by switching to a different camera that only renders the cards on top of the old frame. Each card in turn is given an initial impulse vector in a slightly randomised direction. The 3D scene has bouncy walls around it which leaves the cards bouncing around in a chaotic fashion.

PICTURE 6. The victory animation.

5.2 Data collection

In the data collection mode, the prototype can repeatedly generate new random card configurations and analyse them. The prototype was run on Intel Core i5-4670K processor at 4.0 GHz with the algorithm running in a single thread. With this hardware configuration, a single deal took about 10 milliseconds when the recursive card analysis method is applied to each card. Approximately 100 card configurations can be analysed per second. A simple CSV file output was implemented so that the gathered data could be exported to other programs for further analysis. In this project Microsoft Excel was used to analyse the data.

A game save system for saving and loading the played games was also implemented with the JSON utility from the Unity API. When the game is run for the first time the Example Game 1 is loaded from a game save included in the build. Any new deals and best results for each game are stored in memory. As the game exit is requested the data is saved to the game save in JSON format.

5.3 Results

A large sample of one million random card configurations was generated and analysed. The randomness of the sample was scrutinised by marking down the suit and rank of the first card of each deal. Each suit and rank was expected to appear uniformly in this random sample. In the sample rank three had the highest deviation appearing 0.5% less than expected. Overall, the average deviation from expected distribution was less than 0.2% proving the sample to be sufficiently random.

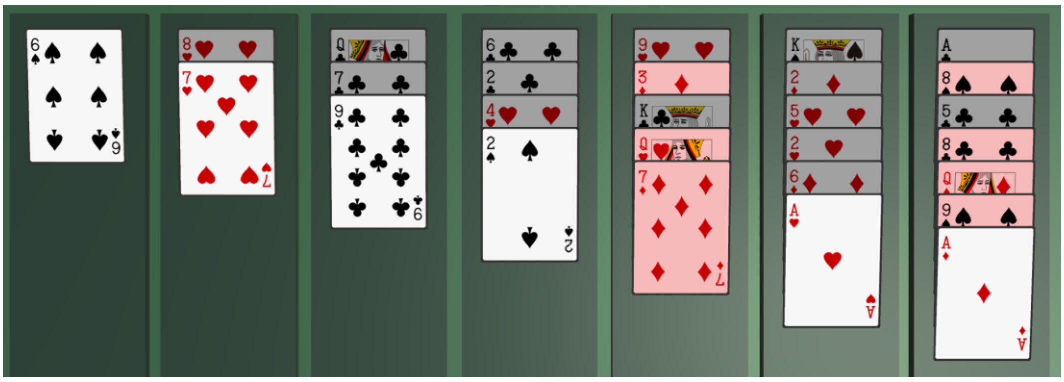
The probabilities for different types of single card solution blocks were compared against the earlier probabilistic percentages from Section 4.3, which were averaged out with a weighted average to reflect all the cards in the game. The bottom cards were excluded from the analysis as they do not block any cards. Table 2 shows that the percentages in the probability calculations and in the random sample correlate strongly. This seems to suggest that the probability calculations correctly predict the outcome and that the random sample was sufficiently random. It would be highly unlikely for both to be faulty in the exact same way.

TABLE 2. Probabilities of different types of solve blocks per card.

Blocked cards	**Expected**	**Observed**
Nothing blocked	67.301 %	67.307 %
Suit blocked	24.071 %	24.056 %
A solver blocked	6.260 %	6.268 %
Solver & suit blocked	2.147 %	2.148 %
Both solvers blocked	0.165 %	0.166 %
Both solvers & suit blocked	0.057 %	0.056 %

As before, only counting the direct solvers gives a simplistic view. An earlier study also attempted to enumerate initial card configurations would force a loss. Yan et al. (2004) could prove 1.19% of the card configurations unsolvable. In this sample 1.16% of deals had one or more unsolvable blockages of this more trivial type. However, blocking direct solvers and suit solution is not the only way an initial card configuration can be unsolvable.

The sample was also analysed with the more advanced recursive method. It was able to detect 2.24% of the games as unsolvable or 1.08 percentage points more than the earlier more trivial analysis. The detection was reviewed manually to verify the results. Besides more complex variants of a card blocking the solutions to itself, a pattern emerged where two cards in different build piles would block the solutions from each other as depicted in Picture 7. This method of recursive analysis was also adopted to run while the game would be played ordinarily. In some specific cases it could detect dead ends, where a card becomes unsolvable when a specific move closes the last remaining solution route permanently.



PICTURE 7. An example of a detected complex unsolvable card configuration.

Earlier experimental results by Blake and Gent showed that at least 81.9% of card configurations are solvable (Blake & Gent 2019, 1–11). Their results used a stricter interpretation of Klondike rules than the rules presented in this thesis. Looser rules should allow for more solvability, but the magnitude of the effect is unknown. However, these stricter results indicate that at most 18.1% of the deals are unsolvable. The recursive method detected 2.24% of deals as unsolvable which means it detected at least 12% of the unsolvable card configurations.

Some playtesting was also conducted where a human test player could achieve win rate of 82% in a sample of 100 random games. These were not first attempt successes, but rather multiple approaches were attempted on the card configurations to maximise win rate. No meaningful connection between the number of different types of blocks and difficulty could be established. Rather the complexity likely arises more from the dependency chains between the cards. Especially the games with the most unintuitive solutions were perceived as the most difficult to solve.

The recursive method was also adopted to run during a game to detect dead ends. The exact conditions when a card becomes permanently unsolvable are quite specific but were encountered occasionally during playtesting. Picture 8 shows one scenario where a non-trivial dead end was detected immediately after the nine of Hearts from Pile 6 was packed on the ten of Spades on Pile 1.

PICTURE 8. An example of a detected dead end.

6 CONCLUSIONS AND DISCUSSION

Historically it was understood that the rules of a solitaire affect the odds of winning. The possible gambling roots help to explain, why winning is so unlikely in the old versions of the game. As Klondike was adopted to contexts outside gambling, new rule variants appeared to offer more difficulty levels. Finally, Microsoft Solitaire transformed Klondike into the casual game it is known as today.

Klondike solitaire is often overlooked or dismissed as a simple game. Instead, scientific scrutiny reveals it is in fact a complex NP-complete problem. There are still many open questions in the field of computer science and especially in computational complexity. Klondike solutions are known to be fast to verify. No efficient way of finding a solution quickly exists yet. Advanced solvers have provided the approximate odds of winning with enough computation time but fail to explain why some games are unsolvable.

A deductive approach was chosen instead to explain why any given card configuration is unsolvable. It seems following the pure Boolean logic of the game rules is too simple for academic analysis, but the simple rules together form a complex whole. Tracing simple card dependencies in depth becomes overwhelming and confusing for human players.

A deeper understanding was gained by developing a playable prototype. The prototype also allowed for the development of further analytics tools and helped to visualise the problem at hand. A card game implementation was found to be a good learning experience as it involved many aspects of programming and game development to fully implement. The game prototype could be developed independently as there were only few external graphical assets.

Based on previous studies and the results of the experiments conducted in this thesis, at least 81.9% of the draw-1 Klondike deals are solvable. At most 18.1% are unsolvable, of which 2.24 percentage points or 12% can be detected with the recursive method that was reported in this thesis. The method offers a fast way to detect some of the unsolvable card configurations and could be used to narrow down the search space of full Klondike solvers. The detection could be developed further to detect even more complex unsolvable card configurations. It could also be adopted to detect dead ends as early as possible in a full solver algorithm.

REFERENCES

Aldous, D. & Diaconis P. 1999. Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem. Bulletin of the American Mathematical Society 36 (1), 413–432.

Bjarnason, R., Tadepalli P. & Fern A. 2007a. Searching Solitaire in Real Time. ICGA Journal 30 (3), 131–142.

Bjarnason, R., Tadepalli P. & Fern A. 2007b. Lower Bounding Klondike Solitaire with Monte-Carlo Planning. International Conference on International Conference on Automated Planning and Scheduling (ICAPS '09) 19 (1), 26–33.

Blake, C. & Gent I. 2019. The Winnability of Klondike Solitaire and Many Other Patience Games. Electronic preprint. arXiv: Artificial Intelligence. https://arxiv.org/abs/1906.12314

Cadogan, A. 1914. Lady Cadogan's Illustrated Games of Solitaire or Patience Revised Edition, including American Games. Revised Edition. Philadelphia. David McKay Company.

Cheney E. 1895. Patience: a series of games with cards. Thirds Edition, with Additions. Boston. Lee and Shepard Publishers.

Demaine E. 2013. Computational Complexity. Lecture. MIT OpenCourseWare. Published on 14.1.2013. Referred on 12.5.2021. https://www.youtube.com/watch?v=moPtwq_cVH8

Dick H. 1896. Dick's Games of Patience or Solitaire with Cards. Second edition. New York. Dick & Fitzgerald, publishers.

Foster R. 1914. Foster's Complete Hoyle: An Encyclopedia of Games. Revised and enlarged edition. New York. Frederick A. Stokes Company.

Great Big Story. Youtube 2017. Meet the Intern Who Wrote Solitaire for Microsoft. Published on 13.4.2017. Referred on 7.3.2021. https://www.youtube.com/watch?v=3x5VAg1HJIg

Hapgood G. 1920. Solitaire and Patience. Edition not stated. Philadelphia. The Penn Publishing Company.

Jensen P. 2020. Celebrating 30 Years of Microsoft Solitaire with Those Oh-So-Familiar Bouncing Cards. Published on 22.5.2020. Read on 15.2.2021. https://news.xbox.com/en-us/2020/05/22/celebrating-30-years-microsoft-solitaire/

Keller M. 2013. What game was played at Canfield's Casino? Published in 2013. Read on 7.3.2021. http://www.solitairelaboratory.com/canfield.html

Kinnunen, J., Taskinen K. & Mäyrä F. 2020. Pelaajabarometri 2020: Pelaamista koronan aikaan. Tampere University. Published on 9.11.2020. Read on 15.2.2021. https://trepo.tuni.fi/handle/10024/123831

Kultima A. 2009. Casual game design values. MindTrek '09: International MindTrek Conference: Everyday Life in the Ubiquitous Era 13 (1), 58–65.

Longpré L. & McKenzie P. 2009. The complexity of Solitaire. Theoretical Computer Science 410 (50), 5252–5260.

Microsoft Studios. Youtube 2017. Microsoft Solitaire Collection: Solvable Decks. Published on 23.2.2017. Referred on 7.3.2021. https://www.youtube.com/watch?v=ZEPm_gliJeI

Morehead A. & Mott-Smith G. 2015. The Complete Book of Solitaire and Patience Games. eBook edition. Google Play Books. Read Books Ltd.

Parlett D. 1980. Solitaire: aces up and 399 other card games. First American edition. New York. Pantheon Books.

Parlett D. n.d. Historic Card Games by David Parlett, Patience. Read on 12.3.2021. https://www.parlettgames.uk/histocs/patience.html

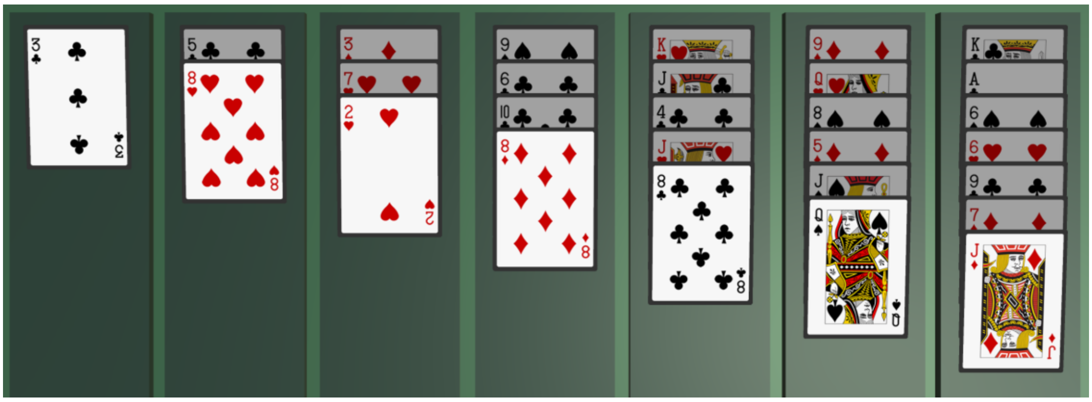
The Strong National Museum of Play. 2019. World Video Game Hall of Fame. Published in 2019. Referred on 7.3.2021. https://www.worldvideogamehalloffame.org/games/microsoft-solitaire

Unity Technologies. 2021. Unity - Scripting API: Random. Website. Published on 15.5.2021. Read on 15.5.2021. https://docs.unity3d.com/ScriptReference/Random.html

Yan X., Diaconis P., Rusmevichientong P. & Van Roy B. 2004. Solitaire: Man versus machine. Advances in Neural Information Processing Systems (NIPS 2004) 17 (1) 1553–1560.

APPENDICES

Appendix 1. Example Game 1 solution



Pile	Instructions	Opens	Prerequisite
3	Found HA through H5	H7	None
7	Pack DJ over SQ	D7	None
7	Pack D7 over C8	C9	DJ solved
4	Pack D8 over C9	C10	D7 solved
4	Pack C10 over DJ	C6	D8 solved
4	Pack C6 over D7	S9	C10 solved
2	Pack H8 over S9	C5	C6 solved
2	Pack D4 from stock to D5		H8 solved
1	Pack C3 over D4	Space	H8 solved
1	Fill the space with DK from the stock		Space
6	Pack SQ over DK	SJ	
6	Pack D10 over SJ		SQ solved
7	Pack C9 over D10	H6	SQ solved
7	Found H6	S6	C9 solved | HA-H4 not blocked
3	Found H7	D3	C9 solved | HA-H6 not blocked
3	Found D3	Space	H6 solved
7	Found SA through S6	CA	H6 solved
7	Found CA	CK	S6 solved
7	Pack DQ over CK from the stock		
6	Pack SJ over DQ	D5	SQ solved
2	Found C3 & D4		
6	Found D5	S8	
6	Found S8	HQ	D5 solved | S7 not blocked
3	Fill the space with SK from the stock		Space
6	Pack HQ over SK	D9	S8 solved
5	Pack C8 over D9	HJ	HQ solved
5	Found H8 through HJ	C4	HA-H10 not blocked
5	Found C4	CJ	HJ solved
5	Pack CJ over HQ	HK	
All closed cards solved			