Internal Report 2010–08 August 2010

Universiteit Leiden
Opleiding Informatica

Klondike strategies

using

Monte Carlo techniques

Pieter Bas Donkersteeg

MASTER THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

LEIDEN UNIVERSITY

Klondike strategies using Monte Carlo techniques

by

Pieter Bas Donkersteeg

A thesis submitted in partial fulfillment for the
degree of MSc Computer Science

in the
Faculty of Science
Leiden Institute of Advanced Computer Science

August 2010

LEIDEN UNIVERSITY

Abstract

Faculty of Science
Leiden Institute of Advanced Computer Science

MSc Computer Science

by Pieter Bas Donkersteeg

Research is done on possible successful strategies in the Klondike card game. A self-made implementation (C++) is used to do the research. One of the techniques used, is Monte Carlo simulation in which a large number of games is played randomly to give the current state a better notion of what move is best.

The odds of winning a Klondike card game are still unknown, which makes it very interesting to get a better understanding of the game, by applying different strategies and see how the game behaves.

In this thesis we emphasize some interesting behaviour of the game in order to come up with hints for a human player to use in their own game.

[Nederlands] Er is onderzoek gedaan naar mogelijk succesvolle strategieën voor het kaartspel Klondike (de bekendste variant van Patience). Voor het onderzoek is een eigen implementatie van het spel gebruikt. Een van de onderzochte technieken is Monte Carlo simulatie, waarmee het spel zich een “idee” vormt van de beste zet op een bepaald moment. Enerzijds worden de resultaten van deze simulaties uitgelegd, anderzijds ligt de nadruk op een aantal interessante aspecten van het spel die onder andere door de simulaties naar voren zijn gekomen. Concluderend wordt onder meer een korte lijst met hints gepresenteerd die het Klondike kaartspel als baken zou kunnen dienen.

Acknowledgements

I would like to thank dr. W.A. Kosters for the valuable discussions, useful comments, and assistance in the creation of this thesis. I like to thank dr. H.J. Hoogeboom for his remarks and help on this thesis as well.

Furthermore, I would like to thank my loved ones for sticking with me through the years.

Contents

Preface

This is the master thesis of my Computer Science Master studies at Leiden University.

To me the most interesting assignments through the whole Computer Science course were the ones concerning games like checkers, chomp and n-queens. I think we programmed and analysed the behaviour of ten or more games in the total Computer Science curriculum.

The reason for this might be the fact that the games are well-known, but the world that opens up when you try to understand their behavior is enormous and full of surprises. It's like a journey into an unknown world (I think).

Klondike is also well-known, but a poorly understood game. Some mathematicians have tried to solve and calculate the odds of winning a game of Klondike, and until today didn't succeed. This makes it a very interesting and challenging subject for this master thesis. It also makes a student humble. Trying to just understand a fragment of the whole problem can be daring enough for one student. And maybe it will motivate me or somebody else to investigate this matter further.

The game of Klondike is very popular under youngsters and elderly. Tens of millions of people around the (western) world play this game every day, whether it is on the kitchen table or on their latest smartphone. The rules of the game are quite simple, but to find a proper solution can be very hard. And chances are that it isn't solvable.

The game of Klondike is played by one person (that is why some Anglo-Saxon countries it is called Solitaire, which means "on your own"). A deck of cards is dealt on the table, some cards remain hidden. The purpose is to sort the deck moving cards from one stack to another following precise rules.

The first part of the work was to make a fast and accurate simulation of the game of Klondike itself. I did just that using the C++ language and made an object oriented card game implementation which plays the game straightforward and reasonably fast. For me this was a grand tour du force, because programming has never been my piece of cake.

During the Artificial Intelligence course I was fascinated by a couple of AI techniques such as Temporal Difference Learning and Monte Carlo Techniques [1]. The story of TD-Gammon [2] and how it changed the way the human "masters" played the game, made me dream. Specifically, I found it interesting to see how a method as simple as Monte Carlo simulation combined with a couple of other Artificial Intelligence techniques could give some insights in the problem of trying to understand how the game of Klondike behaves.

I found it both interesting and challenging to come up with a good game simulator, built from the ground up, and trying to understand the game itself better by using different techniques.

The research into Klondike and Monte Carlo and the writing of this thesis is carried out with the help and continuous support of dr. W.A. Kosters at LIACS, the Computer Science institute as part of the Leiden University. My second supervisor is dr. H.J. Hoogeboom.

Chapter 1

Introduction

Klondike seems to be one of the most popular card games in the world. As it can be played individually, it is a good game to play while spending some idle time. Whether that is with a real pack of cards or on all sorts of devices all over the (western) world. A lot of people ask themselves every time they play this game: why do I end up with a lot of cards still on the table and no move left? Did I take the wrong turn or is this one game just not winnable (or solvable (see Chapter 3.2) as mathematicians call it)? Anecdotal evidence suggests that typical human players win around 15% (approximately 1 in every 7 games) of Klondike games [3]. In this percentage it is very important to know which Klondike version is played, and especially what rules apply. The strictness of the rules determine the outcome of the games to a large extent.

Theoreticians have struggled with this game, referring to the inability to calculate the odds of winning a randomly game as "one of the embarrassments of applied mathematics" [4]. While realtime solvers exist for similar games, no such solver exists for Klondike Solitaire (just yet?).

In this master thesis we are exploring these questions from a variety of perspectives. Which strategies are possible? Can we evaluate a table of cards and say something about the chance of winning? Is there a difference in strategy when playing secured versus transparent Klondike (this will be explained later on)?

Hopefully this master thesis will be a good read and after reading you will pick up your Klondike implementation of choice (again) and try your strategy of choice.

1.1 Patience or solitaire

Solitaire, also called Patience in many countries, often refers to single-player card games involving a layout of cards with a goal of sorting them in some manner. However it is possible to play the same games competitively (often a head to head race) and cooperatively. Patience refers to the fact that winning does not happen too often (and therefore you have to have a lot of patience), where Solitaire refers to the fact that most of the members of the Patience family of games can be played by yourself (alone).

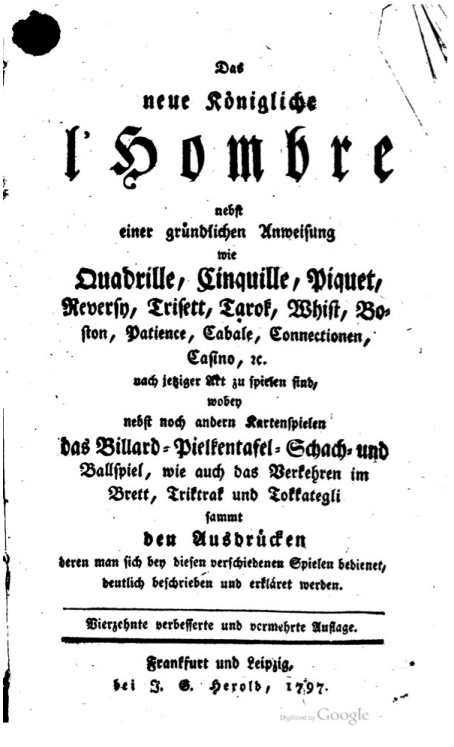
These games typically involve dealing cards from a shuffled deck into a prescribed arrangement on a table, from which the player attempts to reorder the deck by suit and rank through a series of moves transferring cards from one place to another under prescribed restrictions or rules.

There are many different solitaire games, but the term "solitaire" is often used to refer specifically to the most well-known form, called "Klondike". This master thesis will discuss this popular form of Solitaire in detail. See the Wikipedia page with a list of solitaire card games [5] for more information. And [6] is a nice encyclopedia of a lot of different variants of the Solitaire games.

There is a vast array of variations on the solitaire/patience theme, using either one or more decks of cards, with rules of varying complexity and skill levels. Many of these have been converted to electronic form and are available as computer games. Basic forms of Klondike Solitaire and FreeCell come with a lot of different Operating Systems. (Text partially taken from [7].)

1.2 History of the game

Like the origin of playing cards, the origin of Patience is uncertain. The game is most likely German or Scandinavian in origin. The game became popular in France in the early 19th Century reaching England and America in the latter half. Patience was first mentioned in literature shortly after cartomantic (telling the future from the layout of cards) layouts were developed around 1765. The earliest known recording of a game of patience occurred in 1783 in the German game anthology "Das neue Königliche L'Hombre-Spiel" (this book was already published in 1701).



Das
neue Königliche
l'Hombre
nebst
einer gründlichen Anweisung
wie
Quadrille, Cinquille, Piquet,
Reversy, Trisett, Tarok, Whist, Boston, Patience, Cabale, Connectionen,
Casino, ꝛc.
nach jetziger Art zu spielen sind,
wobey
nebst noch andern Kartenspielen
das Billard- Pielkentafel- Schach- und
Ballspiel, wie auch das Verkehren im
Brett, Triktrak und Tokkategli
sammt
den Ausdrücken
deren man sich bey diesen verschiedenen Spielen bedienet,
deutlich beschrieben und erkläret werden.
Vierzehnte verbesserte und vermehrte Auflage.
Frankfurt und Leipzig,
bei J. G. Herold, 1797.
Digitized by Google

r

FIGURE 1.1: The earliest known recording of a game of Patience in *Das neue Königliche L'Hombre-Spiel.*

There is an old tradition in the German or Scandinavian countries to use "Patience" as a guide to what the near future has to offer. This belief assumes that a person's "luck" will vary from time to time and important matters should not be initiated or conducted when the cards are not favourable.

Napoleon was said to have played Patience during his exile; however from written accounts, he played Vingt-Un, Piquet, and Whist but not Patience. The story is thought to have arisen from a misinterpretation. Nonetheless, many solitaire games were named after him, such as Napoleon at St. Helena, Napoleon's Square, etc.

1.3 Klondike

We will discuss and examine the game of Klondike in great detail in the following chapter, but here are some highlights of this popular and addictive member of the Solitaire family. Klondike is the most popular Solitaire variant. Most people,

when asked about the Solitaire game, actually think and talk about the Klondike member of the family.

In the "standard" game of Klondike (of the form: Draw 3, Re-Deal Infinite) all 52 cards (without Jokers) are used; 28 cards are dealt to the tableau (see Figure 2.1) and 24 are put face down on a stock. For a "standard" game of Klondike the number of solvable games (assuming all cards are known) is reported to be between 82 and 91.5 percent [8]. The number of unplayable (not a single move can be done in these games) games is 0.25 percent [9] and the number of games that cannot be won is between 8.5 and 18 percent [8].

1.4 This thesis

In the second chapter of this thesis the game is explained in greater detail. In the third chapter the questions and problem definition are posed.

The fourth chapter will give room to the implementation details of the actual code implementation of the popular game. The fifth chapter will mainly describe the experiment(s).

In Chapter 6 the results will be analysed. Chapter 7, the last chapter, will then give some insights in possible improvements of the discussed implementation. Here also a conclusion about the problem definition is formulated.

Chapter 2

The game explained

In this chapter the different aspects of the game of Klondike are described.

2.1 Klondike

Klondike has become one of the most played computer games, available to hundreds of millions of users worldwide on all major operating systems. For most people it feels like a fairly simple game, but still seems a very hard game to win.

Even theoreticians have been struggling with this game, referring to the inability to calculate the odds of winning a randomly dealt game as one of the embarrassments of applied mathematics. Fact is that there have been published several papers about the odds of the game, but no theoretical basis for it exists just yet.

Klondike is played by one player. The game starts with a specific layout of cards, called a tableau, and the object is to clear the tableau and moving all cards to the four foundation piles as fast as one can.

Taking a standard 52-card deck of playing cards (without Jokers), one card is dealt face up on the left of the playing area, then six downturned or secured cards (from left to right). On top of the downturned cards, an upturned card is dealt on the left-most downturned pile, and downturned cards on the rest until all piles have an upturned card. The piles should look like the ones in Figure 2.1.

The four foundation piles are built up by suit from Ace (lowest card value in this game) to King, and the tableau piles are built down by alternate colours, and

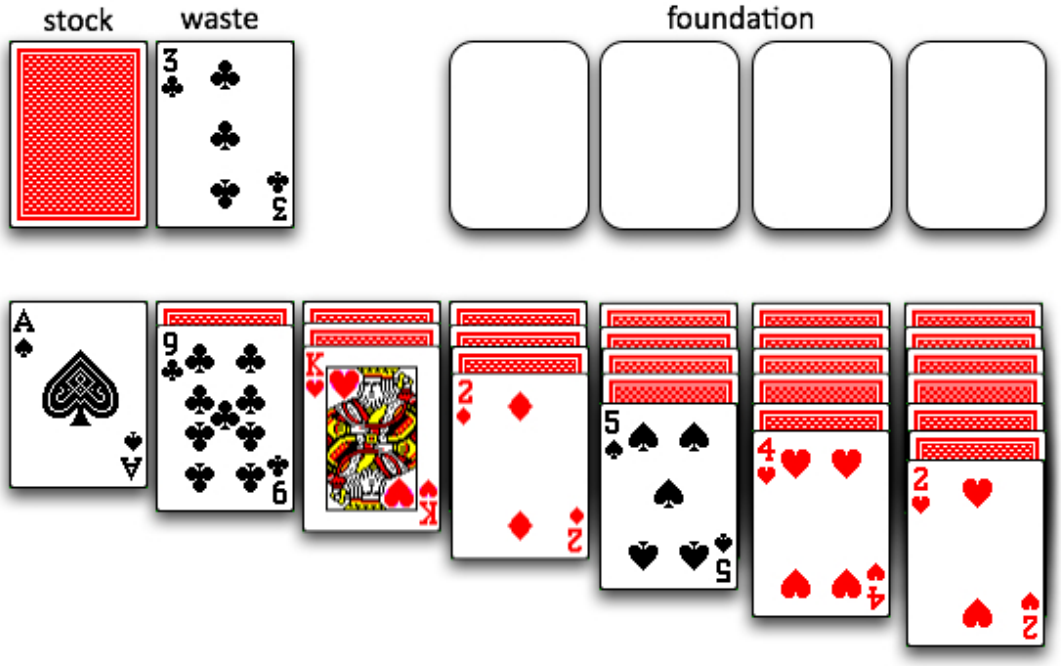


FIGURE 2.1: The table at the beginning of a game of Klondike.

partial or complete piles can be moved if they are built down by alternate colors as well. Any empty pile can be filled (only) with a King or a pile of cards with a King at the top.

There are different ways of dealing the remainder of the deck, each of which can give different outcomes in the actual game:

- Turning three cards at once to the waste, either allowing n passes through the deck or placing no limit on passes through the deck.
- Turning three cards at once, reversing the order of each group of three as the cards are dealt.
- Turning only one card at a time, but only passing through the deck once.
- Turning only one card at a time, but placing no limit on passes through the deck.
- Turning three cards at once to the waste with no limit on passes through the deck, but allowing the player to switch once to a single pass through the deck one card at a time; after that single pass, however, the player cannot go back to turning three cards at a time and can turn over no more cards from the waste.

In the implementation and the experiments that are carried out for this thesis we make use of the variant where three cards are turned repeatedly (no limit to number of passes) and only the topmost card is visible to the player. When this card is moved to the foundation or table, the card underneath comes into play.

Every round the player evaluates the possibles moves that can be played with the open cards on the table and picks the one he/she thinks is the best. This is done every round until one of the following conditions holds:

- all cards on foundation, game is *won*;
- too many stock to waste moves, game is *lost*;
- no moves left (and not all cards are on the foundation), game is *lost*.

These different end states are further explained in Chapter 4.6 on page 27.

2.1.1 Terminology and game talk

In Appendix 1 an extended glossary is included with all the terminology that is used throughout this thesis about the game of Klondike. In this glossary the terminology is explained as it appears in literature and/or used in the text of this thesis. In the next section the terminology in relation to the implementation is explained.

2.1.2 The meaning of things

In literature the different elements of the Klondike game are named differently, so first we want to present a nomenclature to put the terms straight.

Every Klondike configuration consists of thirteen (13) possibly empty stacks. Herewith a list of these stacks is given.

stock (one stack) The *stock* is the stack with all cards faced down and can be used to put new cards on the *waste*.

waste (one stack) The *waste* holds the open cards taken from the *stock*.

foundation stacks (four stacks) These are the stacks that finally must hold all the cards of one suit in order Ace, Two through King.

building stacks (seven stacks) In the standard game there are seven building stacks, together called the *tableau*.

Chapter 3

Questions

One of the central issues in analyzing games, is the issue of evaluating game states. In the game of Klondike, there are two categories of states, the end states in which no more moves can be played and the other category consists of the game states that still have one or more moves to play. In order to say something about the effectiveness of a move, we can evaluate the fitness of its resulting state.

Before we go into the possible fitness values, we want to discuss the three groups of games: unplayable, unsolvable and — of course — the solvable games.

3.1 Unplayable or unsolvable games

3.1.1 Unplayable games

Some randomly generated Klondike games are *unplayable*. This means that no move is possible at the beginning of the game. In the beginning of the game there are 7 open cards (topmost cards of the building stacks on the tableau). The table is unplayable if one turns the stock to the waste eight times ($8 \times 3 = 24$) and none of the reachable stock cards can be played, neither to the foundation nor the tableau.

To let this be the case, three conditions must be satisfied simultaneously [10]:

1. No Aces are in the 7 open cards and 8 reachable stock cards (this happens in 24.4% of the games, see below).

2. None of the open cards on the tableau can be played within the tableau.

3. None of the reachable stock cards can be played to the tableau.

The chance that in all 15 initially reachable cards there are no Aces is:

$$\frac{48}{52} \cdot \frac{47}{51} \cdot \ldots \cdot \frac{34}{38} = \frac{37 \cdot 36 \cdot 35 \cdot 34}{52 \cdot 51 \cdot 50 \cdot 49} \approx 0.244$$

In [10] the probability of occurrence of unplayable games is estimated by Monte Carlo simulation, dealing random moves and checking possible moves. The outcome of this experiment suggests that on average 0.25% of all Klondike games are unplayable. This is a rather small amount of games, and it is more likely that one ends up in an unsolvable game.

Indeed, the number of ways that the seven open cards on the table and the eight "open" cards from the stock can be chosen from a standard 52-card deck is equal to

$$\binom{52}{7} \cdot \binom{45}{8} \;=\; 52!\,/\,7!\;8!\;37! \;=\; 28{,}837{,}689{,}349{,}669{,}200.$$

The number of unplayable configurations among these turns out to be equal to 72,099,595,172,416, or approximately 0.25002%.

This number can be computed by brute force, taking about one minute on a 3 GHz computer. A dedicated C++ program generates all possibilities for red/black cards on the table, without Aces, and without neighbouring cards from opposite colours. For each such configuration, it is relatively easy to count the remaining possibilities for the eight cards in the stock — without generating them in full detail. It is important not to explicitly use the suits, but rather count in terms of red and black.

3.1.2 Unsolvable games

An *unsolvable game* is a game in which, no matter what move you'll do, eventually no more (effective) moves are possible. In some games you still can turn stock to waste, but nothing really changes in the game anymore. It is an area of research to determine whether it is possible to prove if a given (random) table is unsolvable.

The guesses in literature concerning the probability of not solving a game, apart from the unplayable games, are diverse. The article mentioned above guesses somewhere between 2.5 and 10%, whereas in [9] the author guesses that almost 67 percent is solvable, which means a total of 33 percent is unsolvable or unplayable.

One of the main issues in the unsolvable Klondike games is what we call Blocking Card Configurations.

3.1.2.1 Blocking Card Configurations

There are a number of games that cannot be played successfully, because of the way the cards are placed (randomly) on the table.These configurations are called *Blocking Card Configurations* or *Sets of Blocking Cards* In this section some thoughts and statistics on these special card configurations are presented. Just to get an idea about what such card configuration look like, we can ask ourselves what the smallest set of cards is that seems to block a game from being solved. The first configuration we came up with is the one in Figure 3.1, which we think is the shortest set of blocking cards possible in this game.

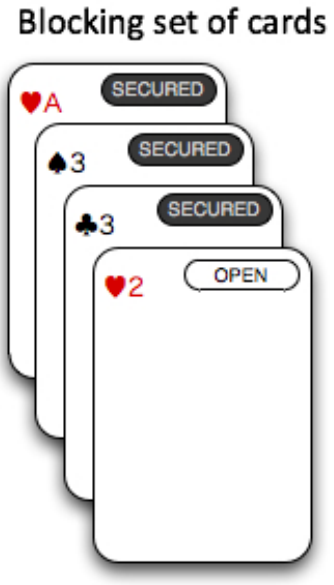


FIGURE 3.1: The shortest example of a set of cards which block a game from being solved.

What really happens is the following. In this type of blocking two conditions are met in combination. The first condition is that the two of hearts (♡2) cannot be

played to the foundation, because the ace of hearts (♡*Ace*) is underneath it *and* (and this is the second condition) the cards that could possibly "help" the ♡2 to escape, namely the ♠3 and the ♣3, are underneath the ♡2 as well. This really blocks a game of Klondike from a successful ending. The order of the three secured cards is not important.

Indeed, we postulate that no sequence of three cards can be a Blocking Card Configuration; This is the case because the three possible destinations of the bottom cards are needed.

We could translate this situation to the longer set of cards in Figure 3.2. The grey starred cards can be any card of any suit or rank.

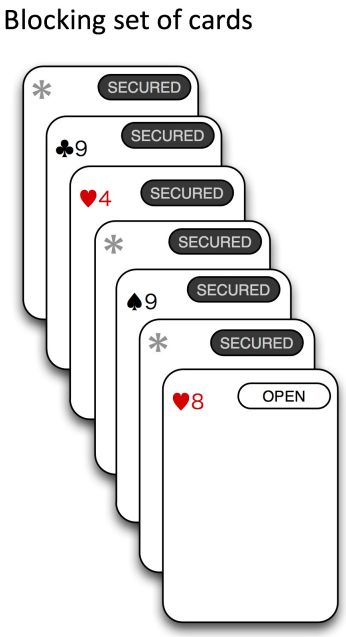


FIGURE 3.2: The longer example of a set of cards which block a game from being solved.

More generally we can say the following about these card configurations:

If card Sr (S being the suit and r being the rank) with color C is open and on top of a table pile (in the examples $Sr = ♡2$ and ♡8, respectively, both with color $C =$ red) and the following cards are secured underneath it, the game cannot be solved. The cards underneath the top cards are: $S'(r+1)$ and $S''(r+1)$ of the opposite colors (in the example S' is ♠ and S'' is ♣, or the other way around) and $Sy : y < r$ with S being of the same suit as the top card. This statement applies to all card configurations, with the top card being $2 \leq r \leq 12$.

A special case in the above statement are Aces. Aces do no have a card $Sy : y < 1$ and therefore are always playable to the foundation and the statement will not apply. Similarly, Kings are also special, since they can be played to empty build stacks.

In this thesis it will not be feasible to come up with all possible blocking sets of cards, but it is quite interesting to do some more research on this matter.

A totally different Blocking Card Configuration is the initial situation where all 15 cards are of even rank.

3.1.3 Odds of blocking sets of cards

The chance we get a Blocking Set of Cards as described in the previous section at the very beginning of a game is calculated here.

The open card at the top can be any card, say x, of any color, say c. Here x should not be 1, i.e., the card should not be an Ace, and also it should not be a King. We now compute the probability that the remaining six cards contain both $(x + 1)$'s from the opposite colors (e.g., if c is ♣, then we mean ♡ and ♢), and at least one card from color c with value lower than x. This last card prevents x from ever going to its foundation.

The probability that six randomly chosen cards from 51 (remember that x of color c has already been chosen) contain two specific ones is

$$(1/51) \times (1/50) \times \binom{6}{2} = 1/85$$

Given that these six cards contain the two specific ones, the probability that they contain at least one card of color c, with value lower than x, is 1 – the probability that these cards contain no card lower than x, of color c. There are $x-1$ forbidden cards. So the probability is

$$1 - ((50 - x)/49) \times ((49 - x)/48) \times ((48 - x)/47) \times ((47 - x)/46)$$

.

We compute this for $x = 2, 3, \ldots, 12 : 0.08, 0.16, \ldots, 0.65$.

All in all the probability for blocking of this kind is approximately $4 \times (1/52) \times (1/85) \times (0.08 + 0.16 + \ldots + 0.65) = 0.0040$, or a chance of 0.40%.

3.2 Solvable

With the percentage of unsolvable games being between 2.5 and 33%, this leaves us with between 66 and 97.5% of solvable games, which is a rather large number of games! These numbers imply that the majority of games are actually winnable, if correctly played. However, experience playing the game indicates otherwise. The reason players do not win most Klondike games is mainly the tremendous amount of guesswork involved in playing Klondike. A few wrong moves can easily make a player lose and this is what happens most of the time.

And therefore it is of crucial importance that we can evaluate the quality of moves with every choice we make. The way to evaluate moves is to look at both the current fitness of the state and the fitness of the states where we are heading. In the next chapter, we will discuss some experiments to come up with a working evaluation function (fitness) that can also help a human player to play the best move possible.

3.3 Fitness of a table

In order to evaluate game states, we have to come up with a good *fitness function* to do so.

For a finished game (see previous chapter) the fitness is normally quite clear. A simple fitness could be to count the number of cards at the foundations. For a type 1 end game (see section 4.6), this means that a score of 52 is the maximum for any game. In Figure 3.3 for example the fitness is $16 = 3 + 5 + 3 + 5$. Also the other two types of end games can be evaluated against this fitness function, but still this means you cannot evaluate tables that are not finished. Secondly we might want the game not only to optimize on the number of cards to the foundation. In this section we discuss some other possibilities.

FIGURE 3.3: An example of a foundation during or at the end of a game.

3.3.1 Moves left

When a game is not in its end state, for example because we want a Monte Carlo simulation not to play every game to its end, but for example some n moves (we call this the depth of a game), we might want to evaluate a state differently.

Ideas we came up with to do so, are the following.

The foundation count should always be a part of the fitness function, because this is the goal of the game. Another possible interesting part of the evaluation could be the ratio between open red and open black cards on the table. This could be a measure for possible combinations in the future. Another interesting property in any state is the ratio of the number of open cards versus the number of secured cards on the table.

Combining some of these ideas into a new fitness function is one of the experiments done in Chapter 4.

3.4 Heuristic strategies

In comparing different strategies, we also want to incorporate heuristic strategies. On the internet there are different sources [11] that try to give a human player certain heuristics to better evaluate states. Herewith we publish a list of common heuristics. The order of the heuristics is not random. The top heuristics are published as more "important" than the bottom ones, as stated on the site. In the following implementation we will make use of some of these heuristics. The heuristics are:

1. Turn up the first card of the stock before making any other moves. It increases the initial number of possible moves and gives you the opportunity to make a better choice.

2. Always move an Ace or a Deuce (or Two) (See Appendix A on page 40) to the foundation whenever it is possible.

3. Always make the play or transfer that frees (or allows a play that frees) a downcard, regardless of any other considerations.

4. Expose secured cards. If you have a choice from several possible moves that expose secured cards, choose the column with the largest number of hidden cards.

5. Hold off the moves that are not important. The best move is one that provides you with the opportunity to make other moves possible or expose secured cards.

6. Do not empty a building stack if you do not have a King to put it in immediately. You gain nothing if you get an empty building stack. A space in Klondike solitaire can only be filled by a King or a sequence starting with a King, so leave your options open.

7. If you have a choice between a black King and a red King to fill a space with, be cautious in your decision. Look at the colour of the blocking cards and make the appropriate colour choice. For example, if you have a red Jack that blocks some hidden cards, you have to select a red King and than wait for a black Queen.

8. Transfer cards from column to column only to allow a secured card to be freed or to make the columns smoother (making them more of equal length).

9. Only play a King that will benefit the column(s) with the biggest pile of secured cards, unless the play of another King will at least allow a transfer that frees a secured card.

10. Only build your Ace stacks (with anything other than an Ace or Deuce) when the play will:

    - not interfere with your *next card protection* (see Appendix A on page 40), or
    - allow a play or transfer that frees (or allows a play that frees) a secured card, or
    - open up a space for a same-color card pile transfer that allows a secured card to be freed, or

- clear a spot for an immediate waiting King (it cannot be to simply clear a spot).

11. Don't play or transfer a 5, 6, 7 or 8 anywhere unless at least one of these situations will apply after the play:

    - it is smooth (building stacks of equal length) with its next highest even/odd partner in the column
    - it will allow a play or transfer that will immediately free a secured card
    - there have not been any other cards already played to the column
    - you have absolutely no other choice to continue playing (this is not a good sign)

12. When you get to a point that you think all of your necessary cards are covered and you just can't get to them, immediately play any cards you can to their appropriate Ace stacks. You may have to rearrange existing piles to allow blocked cards freedom to be able to go to their Ace stack. Hopefully this will clear an existing pile up to the point that you can use an existing pile card with face up to substitute for the necessary covered card.

3.5 Difference in rules

The odds in winning a game of Klondike is greatly influenced by rules that apply. One of the rules we think makes a lot of difference is the number of cards that are dealt with every Stock to Waste move.

3.5.1 Deal-3 versus Deal-1

In Klondike Deal-3 one of the difficulties in the game is that only one third of all cards are accessible directly. The other two third is only available when a card from the deck is played and after the waste is turned to the stock because the stock is empty.

In the current implementation we have chosen for the standard version, in which 3 cards are dealt in one move.

A Human player example shows us that the chance of winning dramatically grows when Deal-1 is played. In one of the experiments we want to see how much a change in these rules can change the odds of winning.

Chapter 4

Implementation and design choices

In this chapter the specific implementation details of the code accompanying this thesis is explained.

4.1 Overview Implementation

The game of Klondike is implemented in a C++ program. This program is an object oriented implementation.

The game follows roughly the following steps in order to determine possible moves and eventually making one of them.

4.1.1 Open Cards

The `OpenCards` function looks at the current table and locates all open cards and puts their position into an array. This list of open cards includes all cards that are turned face up, and therefore parts of open piles are considered as well. Foundation cards are never included in the open cards group, because in this game we do not permit foundation cards to be played back to the table.

4.1.2 Possible Moves

The `PossibleMoves` function looks at all the open cards mentioned in the previous section and summarizes (in an array) all the possible moves (from, to) and its move type. If a move is prohibited, it is not included in this array. Therefore this array holds all possible moves that must be considered playing.

4.1.3 DoMove

The `DoMove` function takes a chosen move (by human hand, random generator or another algorithm, for example the Monte Carlo simulation) and changes the current state to the next state.

Figure 4.1 shows an example of an actual game tree. The `DoMove` function is used to traverse this tree. Note that the bottom right state is reached by two different paths. In the tree representation these paths are different and the state will be generated twice. The figure shows a possibility to combine the two paths at the cost of a more elaborate algorithm.

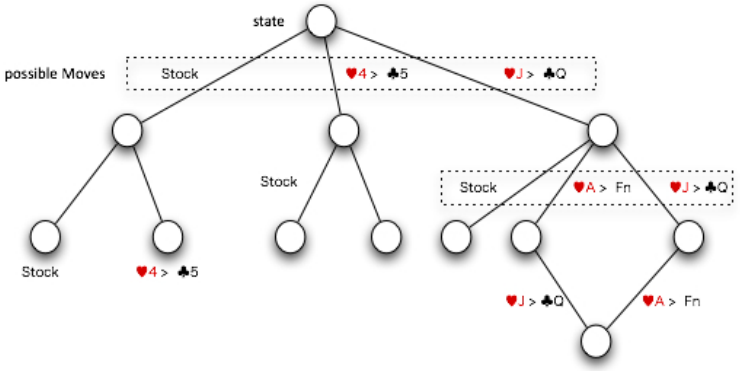


FIGURE 4.1: Representation of the game in a Tree.

4.2 Representation of a Klondike game state

A Klondike game state in this implementation consists of the following elements:

Stock an array holding at most 24 cards, with an index pointing at the topmost, down-turned card

Waste an array holding at most 24 cards, with an index pointing at the topmost, up-turned card

Foundations four array elements each holding the topmost card

Table the table is an array holding the cards at their positions, up or down turned, possibly an empty card

4.3 Representation of a move

A single move is represented by (at least) the following three parts:

From position from this position on the table a move can be done

To position a move can be done to this position

Move type this type holds what sort of move is possible; it helps to deduce the complexity of possible moves; in Table 4.1 one can find all different types possible in the current implementation, see also Figure 4.2.

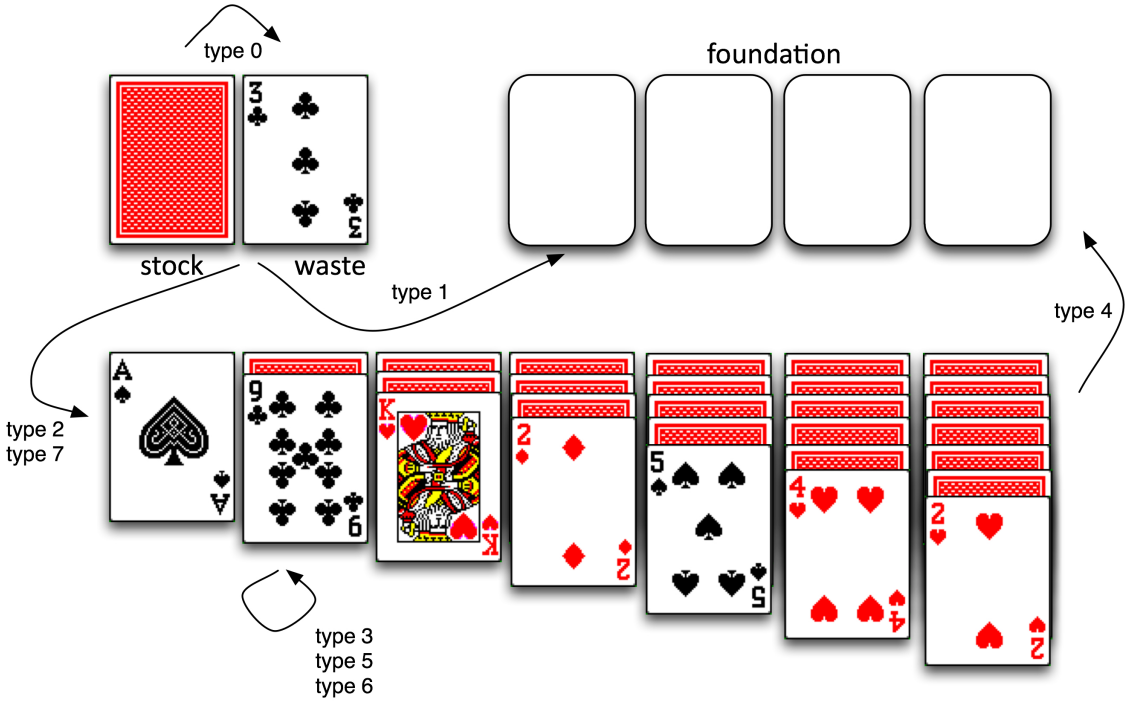


FIGURE 4.2: The different types of moves on the table.

Type	Name	Description
0	Stock to Waste	Turn 3 cards from the stock to the waste stack only presenting the topmost card to the player
1	Waste to Foundation	Turn a single card from the waste directly to the foundation of the correct suit
2	Waste to Table	Turn a single card from the waste anywhere to another card on the table (not being a King, which is another type)
3	Table to Table	Move a single card or a sequence of cards from the table to another column on the table
4	Table to Foundation	Move a single card from the table to the foundation
5	King to Free column	Move a King from the table to a free column
6	Double Move	Move a part of a sequence of cards to another column only if the new topmost card can directly be played to the foundation
7	Waste King to Free Column	Move a King from the waste directly to a free column

TABLE 4.1: Types of moves possible in Klondike.

In the implementation of the game playing a part of an open sequence of cards to another position on the table is valid *only* if the (already) open card just underneath it, can be played to the foundation *directly*. In his presentation P. Diaconis explicitly prohibits such moves (cf. [3]). We currently think that this is no restriction. It might be conceivable that the order within the open cards can be of importance in the later part of the game.

4.4 Introducing Monte Carlo simulation

In the research Monte Carlo techniques are used. In Figure 4.3 one can find a tree representation of the idea.

At a certain point in time T the number of possible moves (in Figure 4.3: 4, T_1', T_2', T_3', and T_4') is more than 1. (When the number of possible moves is equal to 1 we can just do that one move.) We play a defined number of games k in defined

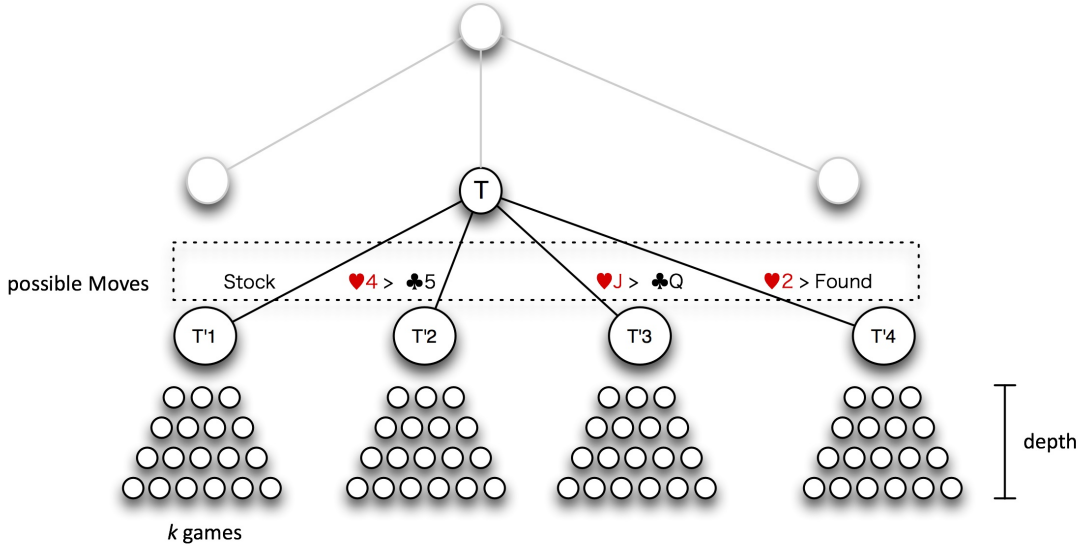


FIGURE 4.3: Tree representation of a Monte Carlo simulation with Klondike.

depth at most d in order to examine the possibility of victory from that point on. We can examine the average score, highest score from all these games, etc. Based on these results, one can decide to take the subtree with best score in order to maximize the chance of success. In case of a tie, a random choice is made between the best scores.

4.5 Monte Carlo simulations

Now that we have an "efficient" Klondike algorithm, we want to examine the game's behavior in order to know what are successful and unsuccessful strategies. In order to do so, an implementation is made in which from any state in the game a Monte Carlo simulation can be carried out to examine possible outcomes from that state on.

The conditions under which the simulation is carried out, are the following:

- number of possible moves ≥ 2;
- there is no rank *Ace* of any suit s that can be played to the foundation;
- there is no rank 2 of any suit s that can be played to the foundation;

- there is no rank k ($k \geq 3$) of a suit s with colour c while all ranks ℓ ($\ell < k$) of the opposite colored suits are on the foundation already; also $k-1$ of suit s has to be on the foundation already.

These are all called safe heuristics, because they do not alter the future of the game in any way, so they are safe to play.

If there is only one possible move, there is no choice and we have to play that one move. If there is an Ace or a 2 you can play to the foundation, it is always the best move. The last case is a bit more complex. If a 3 of any suit s with color c can be played to the foundation and all 2's of the opposite colors are already on the foundation, this card is best to play as well.

4.5.1 Implementation of Monte Carlo

First we make a copy of the current state (Section 4.2) in which the above conditions are checked in advance.

So we end up with a state in which we have two or more possible moves that are of equal fitness, so we have to choose either one of them. Based on these moves we are not able to make a decision what move to do first. In order to make a better decision we carry out a Monte Carlo simulation.

The Monte Carlo simulation is carried out based on some parameters. We can tell the simulation how many games must be played to get an average score. We can ask the simulation not to play all the way to an end position (clear tableau or no more possible moves) or we can even tell the simulation to use a certain strategy to play a game within the simulation (for example: random).

Another parameter can be the way the cards in the game are known to the player. We have defined two variants:

Transparent All cards are known in advance to the player and they are never changed within one game. Not even the cards in the stock.

Secured At the beginning only the topmost card of every column is known. All the other cards are unknown to the player. Herein we define two subvariants:

Shuffled once All 52 cards are shuffled at the beginning of every game and dealt upon the playing table (tableau and stock). After the initial shuffle and dealing, the cards stay where they are until they are turned (opened) into play.

Reshuffle After every move all the secured and therefore unknown cards are shuffled. Alternative: Only the card that is turned is shuffled with the other secured cards.

One of the questions now is whether or not these three variants (Transparent; Secured, shuffled once; Secured, reshuffle) will give different outcomes.

The simulation will play according to the given parameters and will in return come up with a score or ranking per possible move as seen in Figure 4.1.

4.6 The end of the game

A game of Klondike is finished when one of the following is true (cf. Section 2.1 on page 9) :

All cards on foundation, type 1 When all the cards are in the correct sequence (each suit from Ace to King) on the foundation, the game is not only finished, but also *won*.

Too many stock to waste moves, type 2 When the stock is turned to waste too many times (normally twice the stock to waste) and it's the only possible move, no new move will present itself and the game is *lost*.

No moves, type 3 When no possible moves are available, the game is *lost*.

Chapter 5

Experiments

In this thesis a number of experiments are described on the Klondike implementation provided. The experiments aim to maximize the chance of success while playing a move in the game of Klondike.

5.1 Strategies

We have implemented some strategies by means of different players. Each player has a different way of playing the game of Klondike. In this way we can compare the different strategies and evaluate its effectiveness.

5.2 Player characteristics

In the experiments different players with different characteristics are introduced, in order to compare different strategies of the players. Players can be mono strategic, meaning they play by one strategy from begin to end, or multi strategic, meaning that they use different strategies during different parts of the game. We define:

Player H is the human player. It is the implementation in which oneself can play the game.

Player R is the random player. In each and every round it picks a move randomly until an end state is reached.

Player A is the player that chooses randomly based on the highest score with Monte Carlo simulation.

Player C always picks out the heuristic moves first (see Section 3.4), but when no heuristic move exists, it picks again the move with at least one maximum score (based on Monte Carlo).

5.3 Game characteristics

Next to the characteristics of the players, we introduce some game characteristics:

Reshuffle/Shuffle Once Are the secured cards reshuffled just before a secured card is opened? We are interested in the comparison between reshuffled play and play with one shuffle.

5.4 Heuristic ideas

We wanted to outperform the Monte Carlo simulation results, by applying some heuristic ideas. In order to do so, we implemented a system in which points are given to heuristically better moves in order to better predict the best move.

The system receives the moves that, from a Monte Carlo point of view, are all very good moves. They have the same fitness and therefore there is no reason to pick one over the other. Normally we could think we pick one randomly because they are all good moves. But we could also think that with a little more effort we can improve the quality of our decision.

5.5 Fitness function

In the experiments we have used several different fitness functions.

Eventually in the implementation we only used the standard foundation fitness function. This function simply counts the total number of cards on the foundation. For further research it might be interesting to find fitness functions that improve the implementation.

Chapter 6

Results of experiments

6.1 Random

One of the reference experiments was the random playing Klondike game. It picks a random move out of all possible moves at each round.

The experiment holds 100,000 games, and the results are presented in Figure 6.1. The distribution of the fitness of the games is segmented in groups (group 1: 0 points, group 2: 0–5 points, group 3: 6–10 points, and the last groups being group 7: 40–52 points and group 8: 52 points).

It shows that around 11% of all games are actually won in Secured Klondike Random games. The majority of games (84.8%) got "stuck" between 0 and 20 fitness points (number of cards on the foundation).

If one considers the way these Random games progress in time, we get Figure 6.2.

In Figure 6.2 the number of foundation points are found at the vertical axis, and the number of moves is found at the horizontal axis. The different coloured lines are individual games.

In the Random games the fitness really starts to take off after 40 rounds and shows a rather diffuse image. On average a Klondike Random game is won in 113.5 rounds, but the differences in these games are wide. In the Monte Carlo game we will see a different outcome.

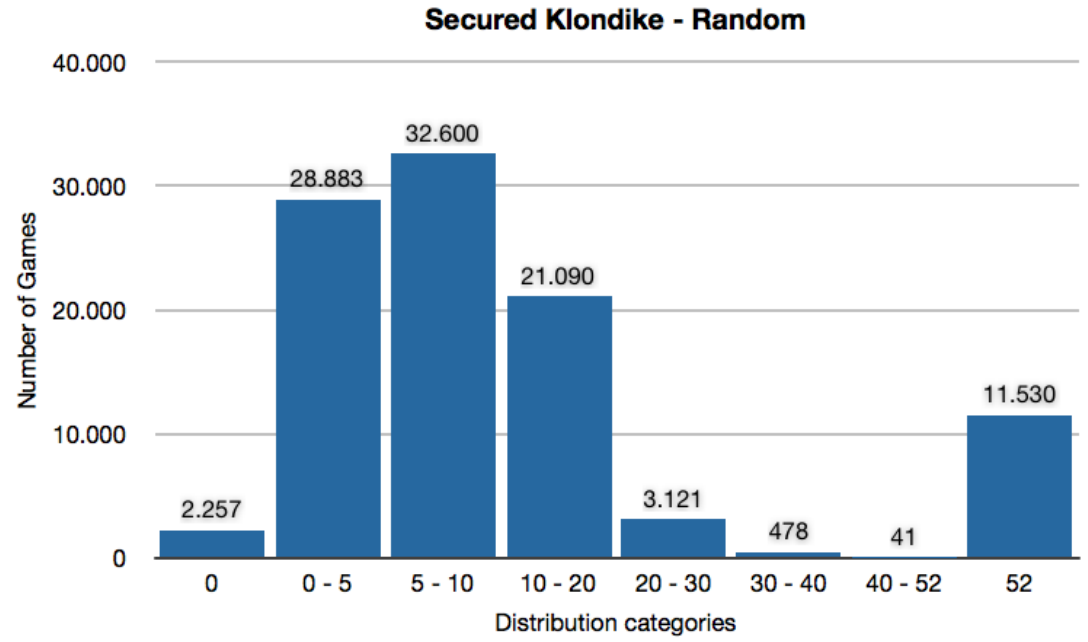


FIGURE 6.1: Distribution of 100,000 Klondike Random games.

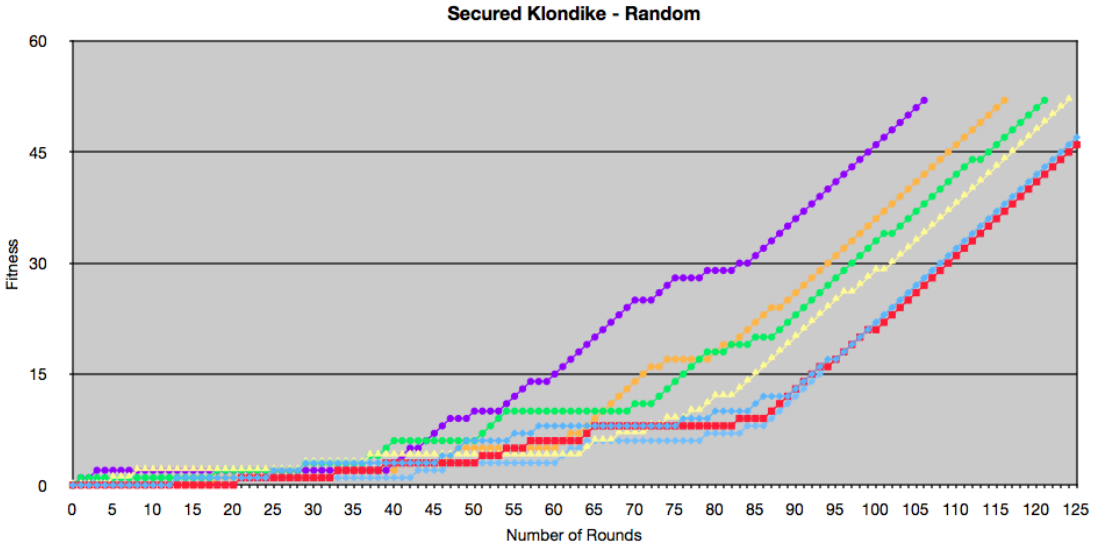


FIGURE 6.2: Path of winning Secured Klondike Random games.

6.2 Monte Carlo

Playing the Secured Klondike game and using the Monte Carlo method to decide between multiple possible moves generates the following results, we arrive at Figure 6.3. For this experiment 10,000 games are played. The percentage going from 11% to 42% looks rather promising.

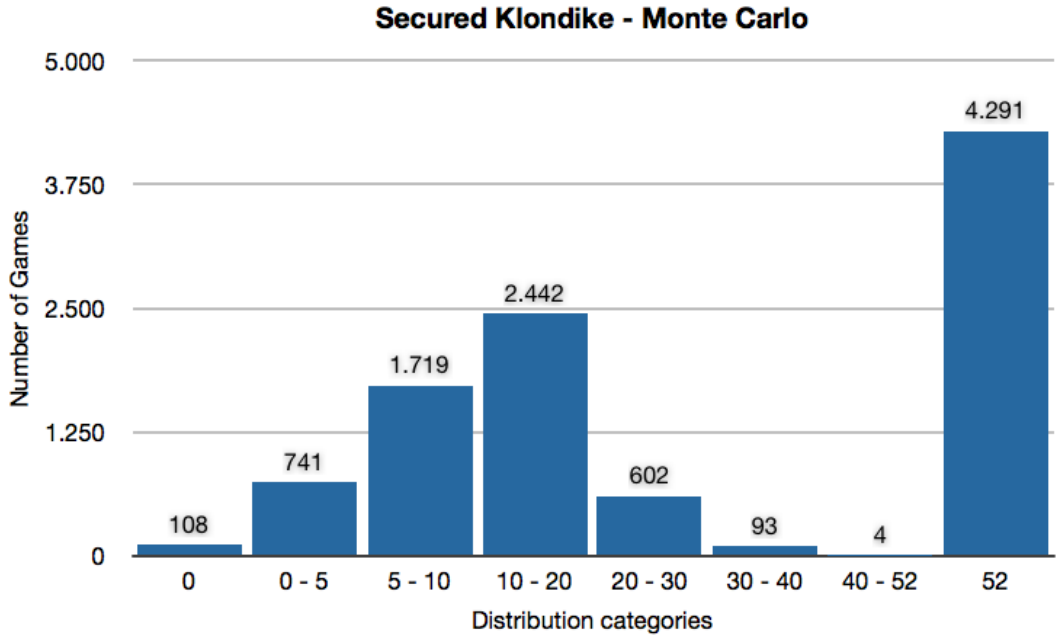


FIGURE 6.3: Distribution of 10,000 Klondike Monte Carlo games, full depth, highest fitness.

A much larger number of games is won (in this example 42.9%, with other experiments we even see 55% of the games solved!) when doing this experiment. Still a larger part (in this example 49.0%) got "stuck" between 0 and 20 fitness points. It would be very interesting to see if heuristics could help these games make even better choices.

In contrast with the Klondike Random games the Monte Carlo games show a much less diffuse image, see Figure 6.4. The fitness takes off right away, and all games grow in time quite linearly. In the results we see that games are won on average in 116.8 rounds. Conclusion is that Monte Carlo games do win more often than Random games, but that it takes a little more rounds to do so. Interesting is whether or not we can win more games and do it in less rounds than Random and Monte Carlo. In the following results we will see just that. We try to do this by using heuristics.

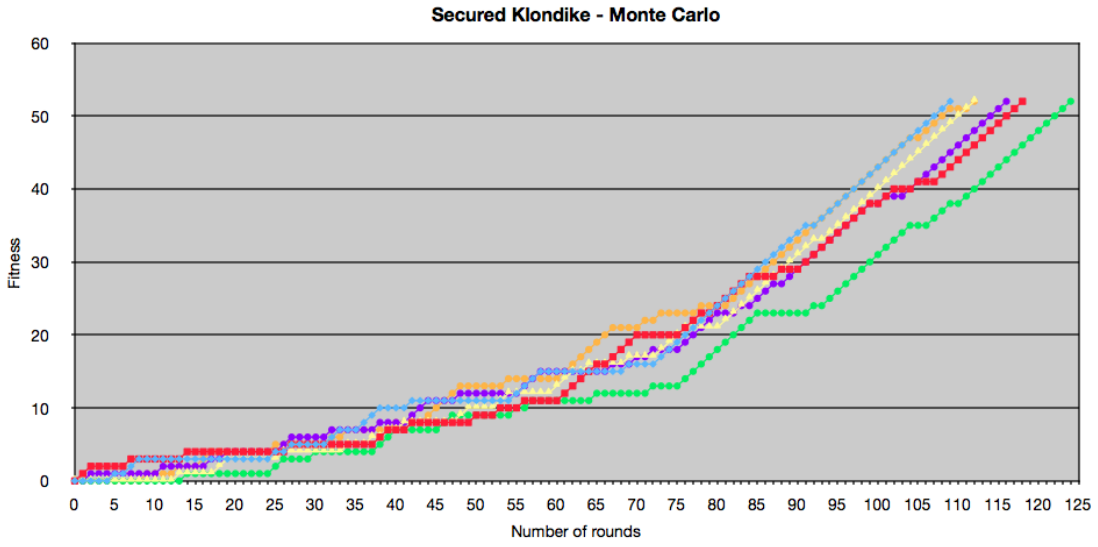


FIGURE 6.4: Path of winning Secured Klondike Monte Carlo games.

6.3 Monte Carlo with heuristics

In this experiment we play the Monte Carlo game, but when we get equal results for the evaluation of the moves at a certain game state, we use heuristics to decide what move to choose. In order to do so, we implemented a system of punish and reward moves based on certain aspects of a game state, such as the number of secured cards underneath an open card or the rank of a movable card in relation to the current state of the foundation.

The reward system is based on the types of the moves (see Chapter 4.3 on page 23). The standard reward is given in order to distinguish between "good" moves and "better" moves. Table 6.1 shows the choices we have made for this function.

The `diffStep` variable is given to try to get a smoother way of playing to the foundation. Not all cards to one colour, but evenly between the colours. In the current implementation we do not want a bigger step than 2 points between the largest and the shortest foundation pile.

In the experiments these values are altered in order to see improvements in the performance, but non was seen during the research.

The number of games won, is currently only slightly higher than the Monte Carlo games, but the number of rounds it takes on average to win a game has dropped

Type of move	Reward	Description
0	3	Standard reward
	20	Aces and Deuces
1	8	Standard reward
	20	Aces and Deuces
	15	Difference on foundation less than `diffStep`
	2	Difference on foundation more than `diffStep`
2	5	Standard reward
3	5	Standard reward
	12	If it opens a secured card
4	8	Standard reward
	15	Aces and Deuces
	15	Difference on foundation less than `diffStep`
	4	Difference on foundation more than `diffStep`
5	3	Standard reward
	12	If it opens a secured card
6	3	Standard reward
	12	If it opens a secured card
7	8	Standard reward

TABLE 6.1: Percentages of games won with different strategies.

to 112.3, which can be seen as a nice result. The results can be found in Figure 6.5 and Figure 6.6.

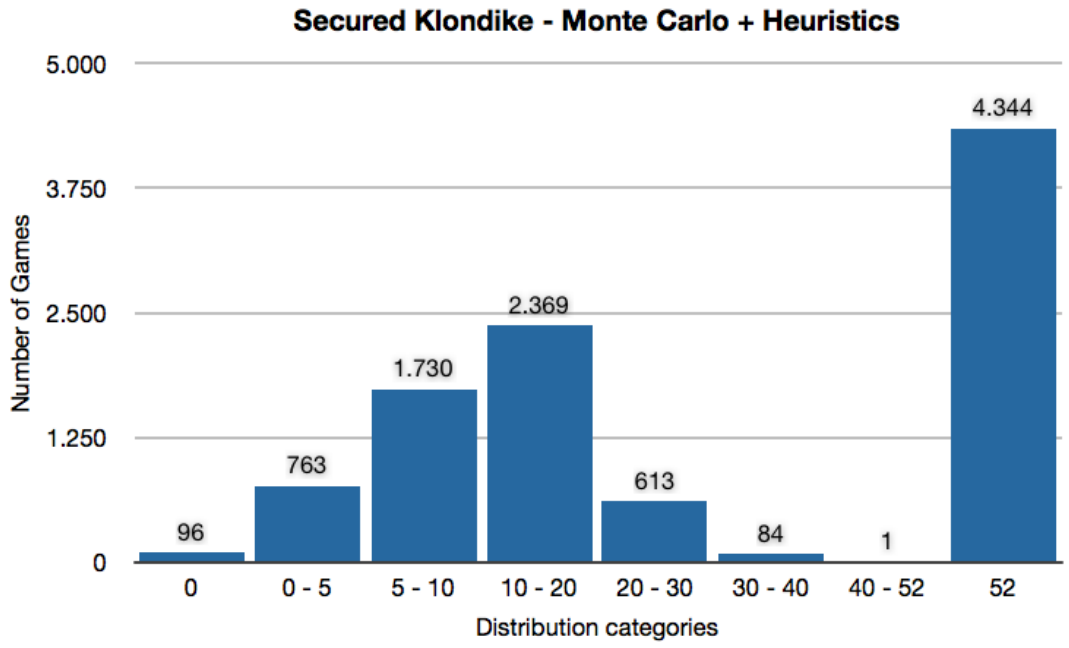


FIGURE 6.5: Distribution of 100,000 Klondike Monte Carlo + heuristics games.

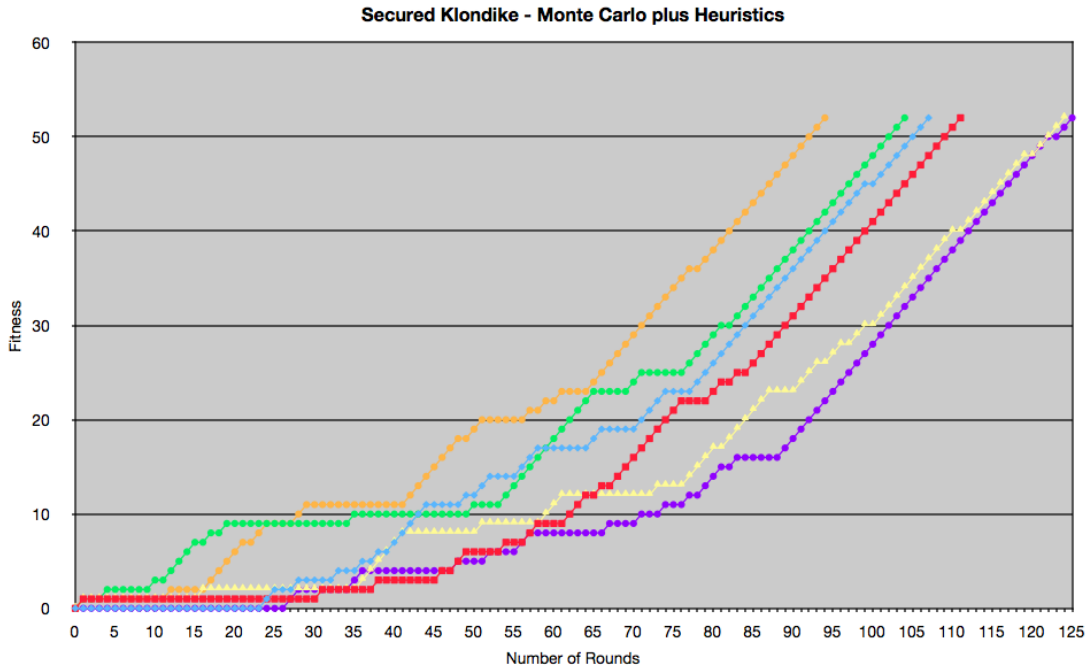


FIGURE 6.6: Path of winning Secured Klondike Monte Carlo + heuristics games.

6.4 Summary of results

If we sum up all the different figures in one table, we see the following results:

	Shuffled Once	Reshuffle	Remarks
Human play	–	10–15%	Own play
Random play	11.8%	11.4%	Fully random
Monte Carlo	43.2%	16.9%	100,000 games
Monte Carlo + heuristics	42.0%	16.7%	100,000 games

TABLE 6.2: Percentages of games won with different strategies.

In the Random simulations we see that almost 12% of the games are won in the "Shuffle Once" setting and 11% in the "Reshuffle" mode. This means that the Random simulation works almost as good in both modes — as expected.

In the Monte Carlo simulation there is a big difference between the two modes. In "Shuffle Once" mode almost three times as many games are won, but the number of won games drops dramatically when played in "Reshuffle" mode. In our opinion the "Shuffle Once" mode can be seen as Klondike with open cards, which is also called Thoughtful Klondike [4] and is explained in the next section. The Monte

Carlo simulation performs only 5% better than the Random simulation in the Reshuffle mode.

In the third strategy we use the Monte Carlo simulation again but try to refine its performance with some heuristic ideas. This idea was nice, but in the results we see no difference.

6.5 Shuffling and Thoughtful Klondike

In the literature research is done on another variant of Klondike, which is named *Thoughtful Klondike* (cf. [4]). In this version of the game the secured cards are also known to the player in advance. The rest of the rules of the game are the same as for Klondike.

If we think about the simulation we have carried out without shuffling unknown cards every once in a while ("Shuffle Once"), we just have implemented the Thoughtful variant of Klondike. The cards in this mode are eventually known to the game and this could be the reason that it performs so well.

The Random implementation performs well in both modes, because it chooses randomly and not a specific move with a specific card.

6.6 Deal-3 vs Deal-1

If we change one of the rules of the implementation we might get very different results. One of the ideas was to change the standard 3 cards deal (from stock to waste) to a 1 card deal. All games in Table 6.3 are played "Secured".

	Deal-3	Deal-1	Remarks
Human play	10–15%	NA	Own play
Random play	11.4%	23.8%	Fully random
Monte Carlo	16.9%	33.3%	10,000 games
Monte Carlo + heuristics	16.7%	34.8%	10,000 games

TABLE 6.3: Percentages of games won compared between Deal-3 and Deal-1.

It becomes clear that in this simulation the results are much better in the Deal-1 than they are in Deal-3 games.

Chapter 7

Conclusions and further research

After doing the experiments in this research it is still quite hard to come up with smarter Strategies than the ones we are already using. The results are somewhat disappointing and do not come close to the percentages given in literature. The game and its behaviour are just very hard to understand due to the complexity. We came up with some interesting thoughts about the unsolvable and unplayable cards.

7.1 Conclusion

Looking at the game and the simulation for so long now, it is still difficult to add some really new ideas. We have tried to come up with some ideas to improve the odds of winning within the boundaries of the possible. Can one, by applying some of the ideas proposed, win 2 out 14 games, instead of the average 1 out 7 for a "normal" human player and become a champion? This seems hard.

When, in a certain sate, a couple of possible moves present itself to the player, and the cards secured are not known, it is very hard to predict what move is best. But what we have learned from the simulation and the large number of games we have played ourselves, we think that the following set of strategy steps can make a player more successful.

1. Turn stock to waste 8×.
2. Aces and Deuces to the foundation.

3. Free column only, if King is directly available.
4. At choice, choose longest secured column first.
5. At choice, leave a gap of only 2 points in foundation.

A master in Klondike is someone with just an inch more luck than the rest of us it seems (for now).

7.2 Further research

7.2.1 Speed and better performance

The code of the implementation is reasonably fast, but for larger experiments with millions of games to be played, the code has to be optimized. The current implementation follows the object orientated practice to a large extent, but for super fast code a less expensive implementation must be more suitable for the job.

7.2.2 Temporal Difference

One of the initial ideas of an interesting thesis subject, was to investigate the possibilities to use Temporal Difference (or TD) techniques with Klondike, just like the TD-Gammon implementation of Tesauro [2]. This was a great idea, but at this moment in time far to ambitious.

One could use the implementation in this thesis to implement a Neural Network to "recognize" certain game states and make decisions based on this. This makes the game more intelligent, something it needs to overcome the flaws in this implementation.

Appendix A

Glossary

card A playing card is built out of a rank and a suit: ♡A is Ace of hearts, ♠J is the jack of spades.

column The tableau on the playing table initially consists of seven (7) columns of cards. The first column holds one (open) card, the second column holds one open card and one secured card, the third column holds one open card and two secured cards in order, until the last (seventh) column that holds six secured in order and one open card.

deck The deck (also called pack) is a complete set of cards in a game deck. A complete set consists of all 13 ranks for all 4 suits ($13 \times 4 = 52$). The deck can be shuffled or not.

deuce Rank two in every Klondike game is sometimes called a Deuce or two Deuces.

foundation The foundation is the part of a Klondike table configuration, where the playing cards eventually are played in order from Ace to King and everything in between.

game A single game of Klondike consists of 3 steps: shuffling the deck, dealing the cards to the table, and moving the cards (serie of moves) until 1. All cards are on the foundation or 2. No possible move is left.

move The smallest part in a game is the move. A move can be either:

1. one or multiple cards from tableau to tableau,

2. turning cards from stock to waste,
3. playing a single card from waste to tableau,
4. playing a single card from waste to foundation,
5. playing a single card from tableau to foundation,
6. playing a single card from tableau to tableau,
7. playing multiple cards from tableau to tableau.

next card protection The fact that you have to make sure that *if* you play a card (lets say ♡ 3) to the foundation that both black deuces are *or* already played to their individual stacks *or* are played to the table.

open card A card is open when it can be played immediately in the current move.

opposite color In the game of Klondike we assume a deck with four ranks and two colors, namely red and black. In the text we call red the opposite color of black and vice versa.

possible moves The possible moves entity holds all the moves that are possible from the current state of the game. Possible moves holds at least one move (stock to waste). If the stock is empty, the waste is turned to the stock and the third card is played to the waste again. This turning to the stock is not count as a move in the current implementation, which is consistent with the rules commonly accepted.

rank The rank of a playing card is one of Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King .

row The tableau on the playing table initially consists of seven (7) rows of cards. Every row holds one card position or placeholder for each column. The row column holds one open and six secured cards, the second row holds one empty placeholder, one open card and five secured card, the third row holds two empty placeholders, one open card and four secured cards, until the last (seventh) row that holds six empty placeholders and one open card.

safe heuristics These are heuristics that can be done without altering the rest of the game in any way. Putting Aces and Deuces to the foundation is the most well-known safe heuristic.

secured card A card is secured when its backside is up and the player cannot see the rank and suit of the card.

stack A *stack* is a pile of playing cards. A stack is called secured when all cards are *faced down* and open when all cards are *faced up*.

stock At the beginning of every game the cards from the deck that are not dealt to the tableau (24 cards) are put on the stock.

suit The four French playing card suits used primarily in the English-speaking world: hearts (♡), spades (♠), diamonds (♢) and clubs (♣). The diamond and heart suits are colored red and the spades and clubs black.

table The table is the state of the game on a certain moment. The table consists of the stock, the waste, the tableau and the foundation.

tableau The tableau is part of the table where the actual playing is done. Cards from the Waste can be moved to the tableau. Cards from the tableau can be played to the foundation. At the beginning of the game 28 cards are laid on the tableau.

waste One of the possible moves in every situation is to move cards from the stock to the waste (unless the stock is empty).

Bibliography

[1] S.J. Russell and P. Norvig. *Artificial Intelligence, A modern approach*. Prentice Hall, 2nd edition, 2003.

[2] G. Tesauro. Temporal Difference Learning and TD-gammon. *Communications of the ACM*, 8:58–68, 1995.

[3] P. Diaconis. The mathematics of solitaire (video), 1999. http://www.uwtv.org/programs/displayevent.aspx?rID=1986&fID=571.

[4] X. Yan, P. Diaconis, P. Rusmevichientong, and B. Van Roy. Solitaire: Man versus machine. In *Advances in Neural Information Processing Systems 17*, 2005.

[5] List of solitaire card games. Wikipedia [retrieved August 25th 2010], 2010. http://en.wikipedia.org/wiki/List_of_solitaire_card_games.

[6] S. A. Barry. *Great Solitaire Games*. Sterling Publishing Co., Inc, New York, 2002.

[7] Solitaire. Wikipedia [retrieved August 25th 2010], 2010. http://en.wikipedia.org/wiki/Solitaire.

[8] R. Bjarnason, P. Tadepalli, and A. Fern. Searching solitaire in real time. *International Computer Games Association Journal*, 30:131–142, 2007.

[9] J. Spolsky. The odds of winning Klondike solitaire. Website [retrieved August 25th 2010], 2007. http://discuss.joelonsoftware.com/default.asp?joel.3.446403.10.

[10] U. Latif. The probability of unplayable solitaire (Klondike) games. Website [retrieved August 25th 2010], 2004. http://www.techuser.net/klondikeprob.html.

[11] J. Yates. Solitaire strategy guide for windows solitaire game. Website [retrieved August 25th 2010], 2009. http://www.chessandpoker.com/solitaire_strategy.html.