

# Klondike Re-imagined

Re-Imagined how?

We're not changing the rules. It's still the same old three-card draw Klondike. But we're letting the computer do all of the mundane stuff. We're introducing some new ways to let you see what is really going on in the game.

- Now the computer calculates all available moves and displays them as *buttons*. To make a given move, push the *button*.
- Instead of making you click on DRAW to see the next triplet in the Talon, now the whole Talon is displayed at once – with *buttons* for all available moves.
- Additionally, now you get to see what the talon will look like the *next time* you go through it – again, with *buttons* for all available moves.
- And if there is a combination of plays that will expose a *downcard* (so you can flip it up), we're going to **hilite** the cards you need to play.

*“Well, if the app takes care of all of this, what's left for me to do?”*

Answer: The fun stuff.

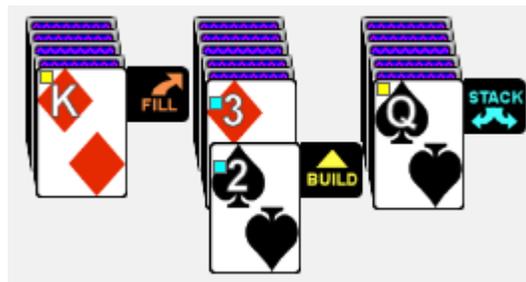
Now, strategies are brought to the foreground: Can you maximize your information? Can you maintain your flexibility? Can you improve the odds that your next flip of a downcard will be useful? And the interesting part is that these strategies are often antagonistic, leading to delicate choices.

And there is also the puzzle aspects; If you need six cards out of the talon, how do you sequence the moves so that you can get all six in the right order? And this is all balanced with stretches of regularity – playing the game can be pleasingly fast.

## A Brief What's New

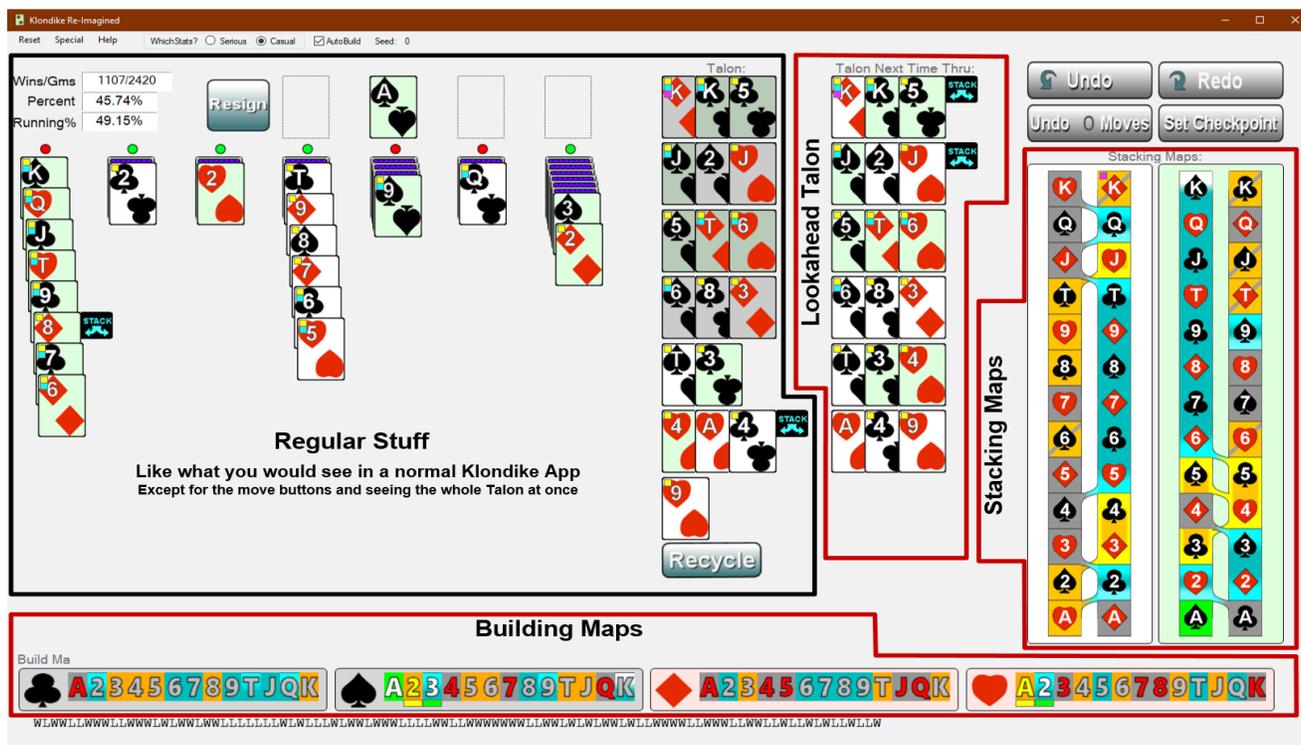
First off, all moves are calculated for you. If a card can be played, a *play-button* is attached to that card. **To play the card, click on the button.**

(Clicking on the card's face causes various hilighting options.)



But that is just the start:

Here is what this re-imagined Klondike looks like. It's got a lot of extra stuff.



All of the regular stuff is there: The “**Tableau**” with its 7 stacks; the “**Foundation**”, the place where you build each suit from Ace to King; and the “**Talon**”, the set of three-card units that you iterate through. We get to see all of the talon at once.

And we have a copy of the Talon. It's called the “**Lookahead Talon**” and it is what the talon will look like the *next time* we go through it. In the above snapshot, we've just played from the 5<sup>th</sup> triplet and the lookahead talon is showing us what moves are coming up in the next iteration. (Notice above that the Lookahead Talon is telling us that we cannot play the 4♣ next time through.)

We have 52 cards to keep track of. So we have two *maps* to tell us *where the cards are* and what their status is:

At the bottom, we have the “**Building Maps**”. All 52 cards are shown there in suit order so you can plan your strategy for building to the foundation.

On the right hand side of the screen, we have a “**Stacking Maps**”. All 52 cards are shown there in *stacking order* so you can plan your strategy for manipulating the tableau's stacks.

Both Maps will show you what cards are the ones that should come into play. If there is a combination of plays that will expose a downcard (so you can flip it up), The maps will **hilite** the cards that need to be played.

Also, the programming has a very sophisticated algorithm for telling when you've lost (so that you don't waste your time on impossible situations and can get on to the next game!).

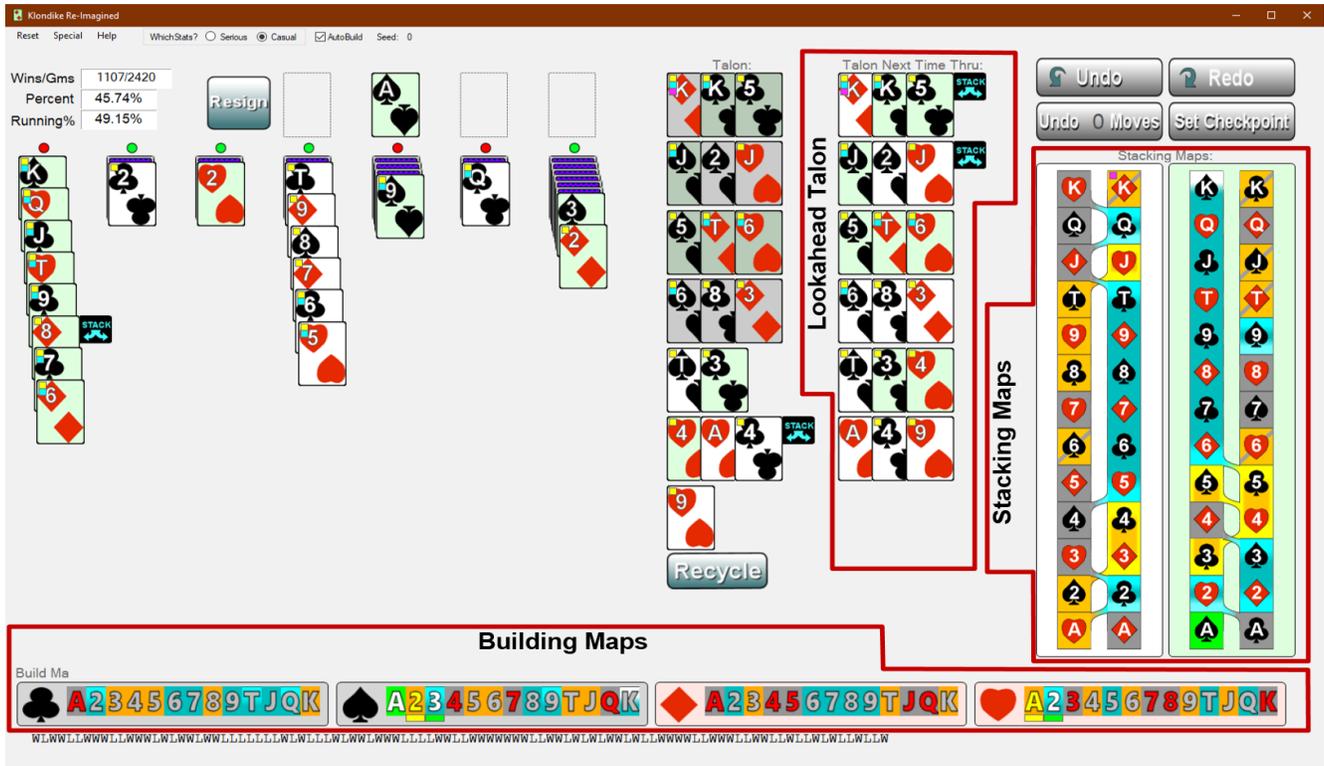
This Klondike is win/loss. You solve the puzzle or you don't (or can't). You are aiming for the highest winning percentage that you can get. With practice, you should be able to win a third to half of your games.

That's it. Go ahead and play a few games. You'll soon get the hang of using the maps.

## The Details

After a bit of playing, you'll notice that there are all sorts of information being displayed visually. You have figured out a lot of them by yourself. But when you get to point of asking, “I wonder what that does?”, This section is for you.

Here are the details. Let's start again with our sample game.



## The Maps

The two maps use color coding so that you can easily recognize where the cards are. The basic color coding is as follows:

Teal	Stacked Cards (face-up in the tableau's stacks)
Gray	Downcards (face-down somewhere in the tableau)
Green	Built (already built to it appropriate pile in the Foundation)
Orange	Talon Cards
Orange with Yellow hilites	"Connector-Cards" These talon cards than can be played to expose a downcard for a <b>flip</b> .
Orange with Light-Yellow hilites	"Connector-Cards" These talon cards than can be played to expose an empty stack to do a <b>king-fill</b> .

## The Building Map

In the the **Building Map**, all 52 cards are shown in *building order* so you can plan your strategy for building to the foundation. For building, each suit is shown separately and the cards are shown in rank order. The basic color scheme is modified to show a bit more information.

The gray downcards are given a *red* rank designation so that they are easy to see. A downcard completely kills the possibility of building any higher card in that suit. And this is reinforced by the color of the other rank

designations. *White* means the card could theoretically built, *gray* means it can't (because a downcard is blocking it). Yellow means that that the given card can be potentially used to expose a downcard or emptying a stack by building.

For example, in our sample deal above, we have a yellow hilited 2♠. This is saying that building the 2♠ onto the already built A♠ would allow the building of the 3♠ which would expose the downcard underneath it, leading to a flip. Notice that in actual practice we would have to move the 2♦ to pull this off.



And in our sample deal, the A♥ is yellow-hilited. If we build the A♥, the 2♥ is buildable and doing so would expose a downcard.



Notice that the 3♠ and the 2♥ in the building map are shown with a top border of light blue. This signifies that these two cards are each covering a downcard.



If it had been a white upper border (like for the K♠), then we would know that the card covers no downcards. That is, if built, would create an empty stack.



And if there is no border, it means that this card sits on another face-up card of the stack.



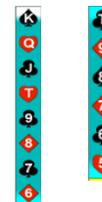
## The Stacking Map

In the the **Stacking Map**, all 52 cards are shown in *stacking order* so you can plan your strategy for manipulating the tableau's stacks.

What do we mean by *stacking order*? The only cards that can go on a black queen are the two red jacks. And in the left-hand half of the stacking map (the "white section"), you see the two black queens with the two red jacks right below them. You may have noticed that half of the cards are "white" and the other half are "green-ish"; and that the stacking map also has a "green-ish section" in tandem with the white section. The stacking rules in Klondike constrain things so that no white card can ever stack on a green card (and visa versa). In terms of stacking, white and green cards are mutually distinct classes. The white cards are the odd red cards and the even black cards. The green cards are the even red cards and the odd black cards.

The Building Map's color scheme is also used in the Stacking Map. But there are some details added to the coloration that provide even more information.

Again, teal is used for face-up cards in the tableau. In addition, all cards in a given stack are connected together as one field of teal. The boundaries of the stack are denoted by a white or light blue border – an upper light blue denotes that the stack sits on a downcard; an upper white border denotes that the stack does not have a down-card underneath it.



Again, orange is the designation for cards in the talon. There are three varieties.

Yellow hilited cards denote a string of "**connector-cards**" - cards that can be played on one of the stacks in such a way that another complete stack can be stacked on it, exposing a downcard for a flip or creating an empty-stack. In our example, the 4♣ can be stacked on the 5♥, then the 3♦ can be stacked on the 4♣, and this allows the 2♠ to be stacked on the 3♦. That exposes a downcard for a flip. And that is exactly what we want to do.



Here is a more complicated set of connector cards. Either black five can be stacked on the 6♦ followed by the 4♥. Now we can stack the 3♠/2♦ on it. Or, we could stack the 3♣ onto the 4♥ so that we could expose the card under the 2♥. The former choice is the better one since we expose the top of 6 downcards rather than a single downcard. Always (almost) choose to expose the bigger stack of downcards!



“Straight” orange cards can't be played right away for setting up a flip. But they are often played to set up something for later .



Orange cards with a gray slash through them can't be stacked. Each requires some downcard to be flipped before it becomes possible to stack them. They are “deadwood”. And if you have too many deadwood cards in the talon, it is bad. They cripple your flexibility.



Again, gray is used for downcards. The gray tells you that the given card is face down in one of the tableau stacks. But of course, there is no way of knowing which stack or which card in the stack until a future flip turns it over.



## How the App Operates

This app automatically takes care of a lot of the mechanics of the game.

All first level move are calculated for you. To carry out a move, just click on the displayed button

If you click on a move button in the lookahead talon, the move is carried out, the lookahead talon becomes the talon, and a new lookahead talon is created.

If **AutoBuild** is checked (recommended), the programming will automatically build tableau cards to the foundation when it determines that there is no point in continuing to leave them in the tableau. This often sets off a chain reaction where lots of cards are moved to the foundation for you – sometimes all of them.

**Automatic Flips:** When you flip a downcard, you lose any accumulated **undos**. So if you have multiple exposed downcards, there is no advantage in not flipping them all. Hence, the programming takes care of that for you: Flipping one exposed downcard flips them all.

The **“Have we lost?” Algorithm:** This implementation has a very sophisticated algorithm for determining a loss. The resign button will light up red (or purple or blue). This algorithm runs in the background; you do not have to let it complete to make another move. You can tell when the loss-algorithm is still executing by the presence of a gray bar between the tableau and the foundation (and the **red/green dots** are hidden). For even more details on this algorithm, see its special section at the end of this document.

**Red dots:** When the downcards are cleared for a given stack, it's an accomplishment. But it's also very bad for your future chances of winning – you have just lost potential for progressing in the game. Why? You used to have **N** piles of downcards to try to expose. Now you have only **N-1** piles to try to work on.

To remind myself of that, I put in the red dots. If there is no chance of exposing a downcard in a given stack, a red dot is placed above it. That includes stacks that are unplayable and, also, stacks that have no downcards. Red dots are not good! If you get 7 red dots, you have lost.

## “Blocked” Markers

The card-faces will show little colored squares when the cards are blocked from certain moves. A yellow-square indicates that the card cannot be built. A blue-square indicates that the card cannot be stacked. A magenta-square indicates that the card was needed for another move but is buried (can't be uncovered) in the talon.



I originally built these as a debugging aid. But I have come to really rely on them for the information they provide.

## Undos

Undos for all non-flip actions are allowed and encouraged. You can experiment to your heart's content – just don't do any flips.

There is also an **Undo N Moves** which undoes clear back to the last flip or to the point where you “set a checkpoint”.



Remember, once you see a new (flipped) card, you can't go back.

## Scoring

Winning and losing is all that matters in this Klondike. This game is a puzzle to be solved. You either solve it or you don't. No partial credit is given.

Four indicators of win/loss are shown:

- The number of games and how many wins
- that percentage
- a Running Percentage (where old results' influence on the score is gradually reduced over time).
- Also, at the bottom of the screen, the last 100 results are displayed.

Wins/Games	803/1860
Percent	43.17%
Running %	42.49%

Seeing the last 100 results at the bottom of the screen is very, very interesting. Obviously, there is a big element of chance in Klondike. And this game can be very streaky. Seeing the last 100 results make the streaks really pop out. If you are in a big losing streak, are you playing badly or is it all just random bad luck?

But then ask the same question about sports teams. When Manchester United loose three games in a row, are they playing badly or is it just bad luck? Hmmmm.

I find that I play with one of two goals:

- to play a quick game or two (maybe waiting for a download), or
- a serious set of games to try and improve my winning percentage.

I definitely don't want my casual games to screw up my winning percentage. So the game keeps **two** sets of results, **Casual** and **Serious**. And for each deal, you can decide which set of stats the result will go in.

Note, you can't change your mind in the middle of a deal.

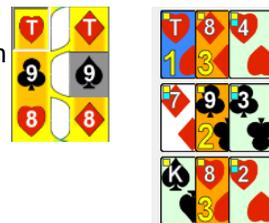
## Hiliting

Clicking on any card's cell in the Building Map hilites that card, its sibling card, the two neighboring build-order cards.

Clicking on any card's cell in the Stacking Map hilites that card, its sibling card, the neighboring stacking-order cards.

Clicking on any card's face hilites that card, its sibling card, the neighboring stacking-order cards and the neighboring build-order cards.

There is one exception to the above hilites: If you left-click on the top card of a set of connector-cards, the programming will hilite the entire set of connector-cards in the talon and the lookahead talon, numbering the pull order and blinking the entire set, one card at a time (in order).



This really helps to visualize how to extract them.

Right Clicking reverts back to the usual hiliting.

## Locks

If you control-click on the face of a card in the Tableau, a **red lock icon** is placed there. The locked 7♠, displayed here, tells the programming that you intend that a particular red 6 goes there and you don't want to accidentally stack the other one there. When you do stack a red 6 on the red-locked 7♠, the programming will stop and ask you if you are stacking the "right one".



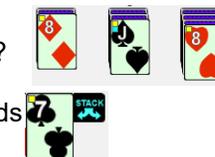
If you control-click on the face of a card in the Talon, a **yellow lock icon** is placed there. The locked T♦, displayed here, tells the programming that you intend that this card should not be moved until a proper moment is achieved. You don't want to accidentally play it too soon. When you try to move it, the programming will stop and ask you if you really want to carry out the move.



## AutoRearrange

AutoRearrange:  ♣♦ ♠♥  ♣♥ ♠♦  off

Suppose we have two red eights in the tableau and we have a 7♣ in the talon with a stack-button attached. When we click on the stack-button, which 8 does the programming choose? It depends on which **AutoRearrange** option we have picked. **Off** means that the 7♣ is stacked on the left-most red 8. The other two options set a default choice of which black cards "go with" which red cards. The first option is clubs and diamonds together and spades and hearts together. So if this option is selected (as shown above) the 7♣ goes on the 8♦. If the second option had been selected, the 7♣ would go on the 8♥.



In either case, if the 7♣ is not where we want it, there will still be a stack-button attached. If we click it, the 7♣ moves to the other red 8.

But there is way more. Lets suppose we have chosen the first option as shown above: clubs and diamonds together and spades and hearts together.

Suppose we have the following stack: If we flip up the 6♠, the programming will *automatically move the 5♥ to the just-flipped 6♠*. This is because we have specified that we want hearts and spades together. Suppose later we flip up the 8♣. Yep, the programming automatically rearranges the 7♦ over to the 8♣.

If at any point the automatic move is disadvantageous, you can manually move it back and it will stay.



The **AutoRearrange** options are found under the **Special** Menu item

## The "Have we lost?" Algorithm – Even more details:

### Talon Extraction Ordering

The have-we-lost algorithm determines whether the cards we need to keep going can be pulled out of the talon. But it doesn't keep track of the order. (Maybe in the future I can tackle that challenge.) This leads to cases where the have-we-lost algorithm does not immediately signal the loss.

Here is such a case: To keep going, we must restack the 5♦, thus emptying a stack for one of the kings. Over in the Stacking Map, we see exactly the (yellow hilited) constructor cards that we need to pull out.



The logic sees that we can pull out the  $T\spadesuit$  and the  $9\diamondsuit$  and figures that we can pull out the  $8\clubsuit$  (since we can move the  $T\spadesuit$ ). But of course, when we play the  $9\diamondsuit$ , the talon recycles, which covers up the  $8\clubsuit$  with the  $6\clubsuit$ .

After a bit, hopefully you'll get enough insight to recognize these cases. In any case, the algorithm will almost always signal the loss on the very next move.

## Red, Purple, and Blue Losses

There are three kinds of losses. The resign button lights up with a different color for each.



If all downcards are covered by cards that are both build-blocked (yellow square) and stack-blocked (blue square), then we have lost. (There is no way to make another flip, so we cannot win.) This condition is checked by the background process.



If we have exposed all of the downcards, but the left hand member of the talon's first triplet is buried-blocked (magenta square), then we have lost. There is no way to build that card, so we cannot win. This condition is checked by the background process.



If there are no (useful) moves, then winning is impossible. This condition is checked by foreground processing. Usually, the background algorithm will later complete and change the loss to a red-loss.

## Special Deals

Under the menu item **Special**, there are two options for playing specific games. Neither option affects your statistics.

### Deal by Seed

Say, you just lost a game and want to replay it. Copy the seed number from the top of the window, click on

**Special.DealBySeedNo** and a dialog will ask for the seed number. Paste it in and hit okay. You can play the deal again (but it doesn't count for your win/loss stats).

### ***Deal by File***

There are two menu options: **Special.File.Load** and **Special.File.Save**. If you are at a crucial decision point, you can save your game and then carry out your decision. If it goes badly, load the file back in and try it the other way. But it won't count.

Also, for some advanced strategies, I plan to set up scenario files where you can load in the file and try your hand.

The files that you save will be located next to your ini.xml file that records your percentages and preferences. The **Help.WhereAml** menu item will show where this is and allow you to go there. Put files that you want to load in this same folder.