# Compact game state representation

Bert Douglas
2023-04-13

## Concept

For any particular type of solitaire game there is a fixed number of groups of cards and a fixed number of cards. Each card must be In some group. There are no cards outside of a group. Some groups may initially be empty or become empty during play. Some empty groups may also become non-empty during play.

A group is represented by a group-byte followed by zero or more card-bytes comprising the group. The entire game state is represented by a simple concatenation of all the groups.

During game play cards can move between groups. However, these properties remain invariant:

- ordering of groups
- number of cards
- number of groups
- number of bytes used for each game state

The ordering of groups is typically used (elsewhere) to determine the role of each group, such as foundation, tableau, etc. However, this data structure contains no information regarding roles of groups.

## Encoding of the Group Byte

| Group Byte | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bit position** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Value or Use** | **1** | **Number of cards in group [0..127]** | | | | | | |

## Encoding of the Card Byte

| Card Byte | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bit position** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Value or Use** | **0** | **FaceUp** | **Suit** | | **Rank** | | | |

| FaceUp | |
|---|---|
| **Bit Value** | **Meaning** |
| 0 | Card is face down and not visible |
| 1 | Card is face up and visible |

| Rank | | | |
|---|---|---|---|
| **Bit Value** | **Hex Value** | **Rank Full Name** | **Rank Abbreviation** |
| 0b0000 | 0x0 | Ace | A |
| 0b0001 | 0x1 | Two | N2 |
| 0b0010 | 0x2 | Three | N3 |
| 0b0011 | 0x3 | Four | N4 |
| 0b0100 | 0x4 | Five | N5 |
| 0b0101 | 0x5 | Six | N6 |
| 0b0110 | 0x6 | Seven | N7 |
| 0b0111 | 0x7 | Eight | N8 |
| 0b1000 | 0x8 | Nine | N9 |
| 0b1001 | 0x9 | Ten | N10 |
| 0b1010 | 0xA | Jack | J |
| 0b1011 | 0xB | Knight/Cavalier | C |
| 0b1100 | 0xC | Queen | Q |
| ~~0b1101~~ | 0xD | King | K |
| 0b1110 | 0xE | (reserved-R) | R |
| 0b1111 | 0xF | (reserved-S) | S |

| Suit | | | | |
|---|---|---|---|---|
| **Card Byte** | **Suit** | **Round Top** | **Red** | **Unicode** |
| 0b00 | Spades | 0 | 0 | 0xA0 |
| 0b01 | Diamonds | 0 | 1 | 0xC0 |
| 0b10 | Clubs | 1 | 0 | 0xD0 |
| 0b11 | Hearts | 1 | 1 | 0xB0 |

## Relation to Unicode

The assignment of codes for Rank copies the values used in Unicode. See: https://en.wikipedia.org/wiki/Playing_cards_in_Unicode. Unicode is notable for having a card with rank of Knight/Cavalier in addition to the usual Jack, Queen, King. The Knight rank is reported to be used in Spain and Italy, instead of the Queen. Without much study, it seems wise to follow Unicode values for rank of cards.

Sadly the Unicode values for Suit are not convenient to use internally. Instead a different encoding is used, where one bit indicates if the card is black or red. The other bit has been whimsically chosen to indicate if the Suit symbol is pointed or rounded on the top. These two attributes happen to be orthogonal. This requires a small translation table to convert from Unicode to internal code.

The Unicode value (U) for a card may be calculated from the rank value (R) and the suit value (S) by:

```
U = 0x1F001 + ({A,C,D,B}[S]<<4) + R;
```

## Estimation of memory usage

For Klondike there are 13 groups and 52 cards, for a total of 65 bytes per state. If it takes 200 moves to win, then this is 13,000 bytes. In today's computers, this is nearly insignificant. In one megabyte you could represent about 7700 states.