Unicode Magic Tricks 🪄🎩

Nicolas Hurtubise
DIRO
Université de Montréal
Montréal, Canada
nicolas.hurtubise at umontreal.ca

***Abstract*—Pretty much what you could expect from a paper that contains emojis in the title.**

***Index Terms*—Unicode, magic trick, emojis, bitwise operators, sleight of bits**

I. Introduction

As of April 8 2020, according to a survey realized during the COVID-19 pandemic, approximately 50%[1] of the adult population started to learn magic tricks as a way to pass the time during lockdown. This is an odd decision, as the close-up magic tricks aren't really compatible with the idea of socially distancing. Some magicians tried to adapt their acts by doing video-conference tricks, but the combination of limited bandwidth, low frame rates and dropped frames all tend to degrade the magical effects. A possible solution lies in the world of text conversations.

In 2010, the standard 52-cards deck was introduced to the emoji world (Figure 1) as part of Unicode 6.0, using the range of code points from U+1F0A1 to U+1F0DE [1]. This opens the door to a variety of new card tricks, which could be performed 100% digitally, even on horribly slow internet connections.

This paper describes a few of the possible magic tricks that could be performed entirely using Unicode emojis. The concept of *sleight of bits* is introduced as a technique to turn a Unicode code point into another one, while looking as if nothing suspect happened.

II. Magic tricks

A. Color change

Description: In this trick, a card is selected by an audience member. The magician then changes its color in front of everyone's astonished eyes. A red card is turned into a black card, and vice versa, while preserving the same rank.

Method: While this method could be achieved in various ways, the key to good *sleight of bits* is to change as few bits as possible.

For a given card, the corresponding binary code point can be dissected into three parts:

- *Bits 7–31:* playing card prefix, identical for every card

[1]That was a home-made survey. I actually socially distanced during this time and the only person I met at home was my roommate. He didn't start doing magic tricks, but I did.

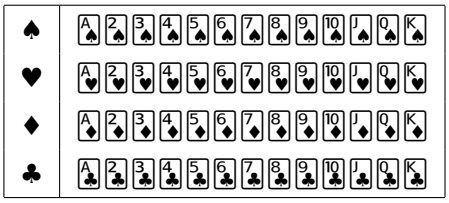

Fig. 1. Standard 52-cards deck in Unicode symbols [2]

- *Bits 4–6:* suit bits
- *Bits 0–3:* rank bits

For a given suit, bits 0 to 3 can be set to change the rank. For a given rank, bits 4 to 6 can be changed to set the suit:

Emoji	Code point (binary)
Same suit	
🂡	0...0001111100001010<u>0001</u>
🂪	0...0001111100001010<u>1010</u>
🂮	0...0001111100001010<u>1110</u>
Same rank	
🂮	0...0001111100001<u>010</u>1110
🂾	0...0001111100001<u>011</u>1110
🃎	0...0001111100001<u>100</u>1110
🃞	0...0001111100001<u>101</u>1110

To perform the color change, use *sleight of bits* to quickly flip the fifth and sixth bits, to obtain the conversion

♠ 010 ↔ ♦ 100 ♥ 011 ↔ ♣ 101

This can be achieved in your favorite language using operators such as

```
code_point =
    (code_point & 0xFFFFFF9F)
    | (~(code_point & ~0xFFFFFF9F))
    & ~0xFFFFFF9F;
```

This will effectively change black cards into red cards, while retaining the card's rank.

As a misdirection, you can always recite the famous hexamagical incantation :

0xABACADABA

which will give more credibility to your act.

B. Card vanish

Description: In this trick, a card is selected from an audience-shuffled deck and the magician makes it disappear from the program.

Method: This method relies on a few key components, and requires a small setup beforehand. Even though the deck should be audience-shuffled, the selected card is actually forced through the selection of an appropriate random seed beforehand. Let an audience member call the `shuffle` function, but make sure the current seed will result in the King of Spades being the top card.

Once the pack is shuffled, show the first card to the audience. Using sleight of bits, change the least significant bit of its code point to a 1:

```
code_point |= 1;
```

The resulting code point, U+1F0AF, is not assigned as of Unicode 13.0 [1]. The card will thus look like it has vanished into an undefined character.

You can expect your audience to look a bit like this:

C. Metamorphosis

Description: In this trick, the magician lets an audience member pick a card from a shuffled deck, then turns it into a dove 🕊.

Method: As with the last trick, the metamorphosis is best done with a forced card. Using the 10 of Diamonds leads to the smallest hamming distance between binary codes, making the sleight of bits slightly more convincing:

Emoji	Code point (binary)
🃊	0...0111110<u>0</u>0<u>1</u>1001010
🕊	0...0111110<u>1</u>0<u>1</u>01001010

Force the 10 of Diamonds to be chosen, and show it to your audience. As they closely inspect the card to tell whether it's a gimmicked emoji or a real one, quickly flip the bits 7, 8 and 10, as such:

```
code_point =
    (code_point & 0xFFFFFA7F)
    | (~(code_point & ~0xFFFFFA7F))
    & ~0xFFFFFA7F;
```

This effect, when done properly, is truly stunning.

D. Mind-bending

Description: For this trick, an audience member thinks of any card and writes it down in secret as a `const` value, so that it cannot be changed later. The magician declares to have divination powers that allows them to always correctly determine which card was selected. The magician then guesses the *wrong card*. The audience member proves it by turning around the card, which is revealed to have changed to become the magician's guess.

Method: This card trick is better implemented in C. Make an audience member write down any card they can think of as a `const` value, after signing it.

```c
// mind-bending.c
#include <stdint.h>
#include "stdio.h"

int main() {
  // Your card here, as a utf-8
  // sequence, e.g. Ace of Spades
  const uint64_t
  code_point = 0xAA1829FF0;

  FILE* out = fopen("reveal.txt", "w");

  fwrite(&code_point,
      sizeof(uint8_t), 5, out);
  fclose(out);

  printf("I predict... ");
  printf("The 3 of Diamonds!\n");
  printf("0xABACADABA!\n");

  return 0;
}
```

Upon inspection by a spectator, this code looks quite innocent. The deceptive part lies in the inclusion of the *local file* `"stdio.h"` instead of the usual `<stdio.h>`. This file is the one doing all of the heavy-lifting:

```c
// stdio.h
#include <stdio.h>
// Swap for the 3 of Diamonds
#define FILE \
  uint64_t $;*(&$+1)=0xA83839FF0;FILE
```

The key to this trick is that the `FILE` type is actually redefined as a macro that expands in a sleight of bits over an overflowed address containing the "constant" value. Some might argue that the C language itself is the strongest misdirection at play here.

This effect can be rendered even stronger by allowing the audience member to *sign the chosen card* first. A signed card can of course be obtained by using any combination of *Combining Diacritical Marks*, for instance 🃃⃝ or 🃊⃝.

III. Conclusion

This paper proposed a very niche joke that's targeted at people who are *both* computer programmers and magicians. That's not a lot of people. If Alex Elmsley was still alive, he would probably slightly smile and then move on to work either on actual magic tricks or useful computer programs.

On second thoughts, maybe you should not publish this.

References

[1] "Playing Cards, Range: 1F0A0–1F0FF", The Unicode Standard, Version 13.0

[2] "Unicode character database", The Unicode Standard (online)