ESP/Ass3 WS18

Aus Progwiki
< ESP

Inhaltsverzeichnis

Lernziel

Basic I/O, Funktionen, Datenstruktur: Structs, Linked Lists, Speichermanagement (malloc, realloc..), Rekursion

Kartenspiel

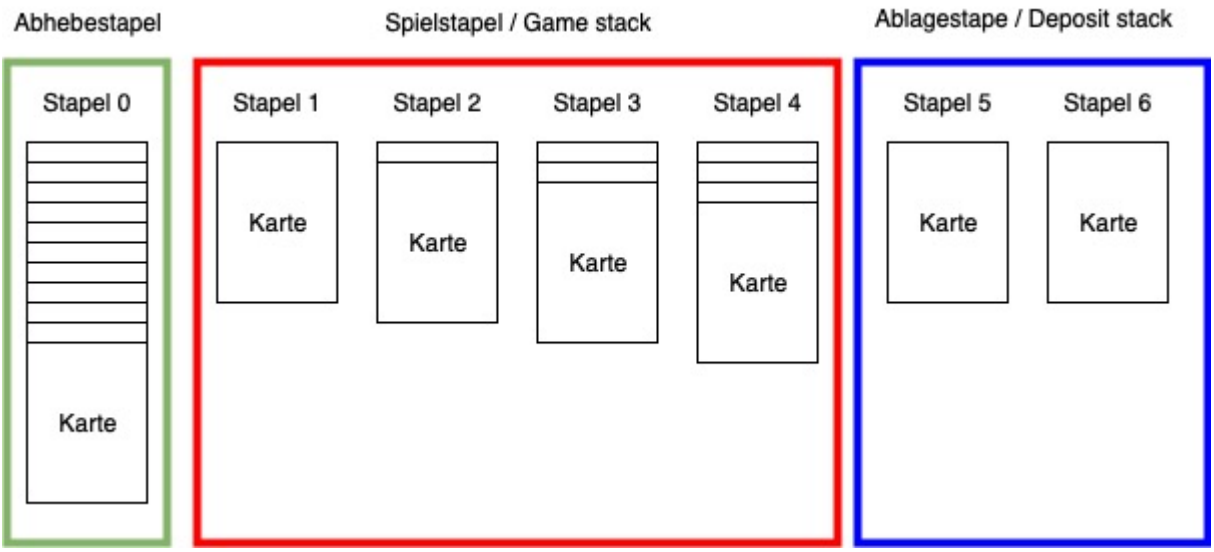
Im dritten Assignment werdet ihr ein vereinfachtes Solitär Kartenspiel implementieren. Dazu muss das Programm die unter aufgelistete Aufgaben erledigen können.

Für die Umsetzung des Spieles müsst ihr mit "Single-Linked-List" bzw. "Double-Linked-Lists" arbeiten und diese dynamisch allokieren. Dafür solltet ihr mit **malloc** für das Reservieren und mit **free** für das Freigeben von Speicher arbeiten. Wie schon im letzen Assignment soll das Programm bei EOF beendet werden.

Info

- Das Einlesen der Konfigurationsdatei erfolgt case-sensitive
- Das Einlesen bzw. Verarbeiten des Userinputs verläuft case-insensitive
- Der Userinput kann darüberhinaus eine beliebige Länge haben

Spezifikationen



Stapel

Es gibt insgesamt sieben Stapel im ganzen Spiel, welche in folgende Arten unterteilt werden:

- 1x Abhebestapel
- 4x Spielstapel (Game stack)
- 2x Ablagestapel (Deposit stack)

Karten

Es gibt zwei verschiedene Farben (color), von jeder Farbe (color) gibt es jeweils 13 Karten, wobei jede Karte einen anderen Kartenwert (value) besitzt, also gibt es insgesamt 26 Karten im Spiel!

Es gibt dreizehn verschiedene Kartenwerte (value)

- A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K

Es gibt zwei verschiedene Kartenfarben (color)

- rot (red), schwarz (black)

Spielregeln

[18.12] Das **ZURÜCKLEGEN** von Karten auf den **ABHEBESTAPEL** ist nicht erlaubt, da dieser nur zum **ABHEBEN** verwendet werden soll.

Verschieben von Karten auf einen Spielstapel (Game stack)

Das Bewegen von Karten auf den 4 Spielstapeln (1, 2, 3, 4) soll wie im echten Solitär erfolgen, d.h man kann eine einzelne Karte bewegen oder einen Stapel an Karten. Ein Bewegen eines Stapels an Karten ist jedoch nur erlaubt wenn dieser sich schon in der richtig sortierten Reihenfolge befindet.

Beachtung der Kartenfarbe

Eine Karte darf nur auf einen Spielstapel verschoben werden, wenn die Farbe (color) der zu verschiebenden Karte nicht die selbe ist, wie die Farbe der Karte, welche am Ablagestapel liegt.

- z.B: Karte mit der Farbe (color) rot (red) **darf** auf die Karte mit der Farbe (color) schwarz (black) gelegt werden.
- z.B: Karte mit der Farbe (color) rot (red) **darf nicht** auf die Karte mit der Farbe (color) rot (red) gelegt werden.

Beachtung des Kartenwertes

Eine Karte darf nur auf einen Spielstapel verschoben werden, wenn die zu verschiebende Karte einen um eins geringeren Kartenwert (value) gegenüber der obersten Karte des Ablagestapels hat.

- z.B: Karte mit dem Wert (value) 2 **darf** auf die Karte mit dem Wert (value) 3 gelegt werden.
- z.B: Karte mit dem Wert (value) 3 **darf nicht** auf die Karte mit dem Wert (value) 3 gelegt werden.

Verschieben von Karten auf einen Ablagestapel (Deposit stack)

Die Ablagestapel sind zu Spielbeginn immer leer und werden im Laufe des Spieles gefüllt.

Sollte sich noch keine Karte auf einem Ablagestapel befinden ist die erste Karte die darauf gelegt werden darf immer ein Ass, da dies die niedrigste Karte des Spieles ist!

Beachtung der Kartenfarbe

Eine Karte darf nur auf einen Ablagestapel verschoben werden, wenn die zu verschiebende Karte einen um eins erhöhten Kartenwert (value) gegenüber der obersten Karte des Ablagestapels hat.

- z.B: Am Ablagestapel liegt die Karte (schwarz, A) und wir wollen die Karte (schwarz, 2) darauf legen. Das ist **erlaubt**, da die Karte dieselbe Farbe (color) hat.
- z.B: Am Ablagestapel liegt die Karte (schwarz, A) und wir wollen die Karte (rot, 2) darauf legen. Das ist **nicht erlaubt**, da die Karte nicht dieselbe Farbe (color) hat.

Beachtung des Kartenwertes

Eine Karte darf nur auf einen Ablagestapel verschoben werden, wenn die Farbe (color) der zu verschiebenden Karte die selbe ist, wie die Farbe der Karte, und der Kartenwert um eins höher ist als die Karte, welche am Ablagestapel liegt.

- z.B: Am Ablagestapel liegt die Karte (schwarz, A) und wir wollen die Karte (schwarz, 2) darauf legen. Das ist **erlaubt**, da die Karte der direkte Nachfolger ist.
- z.B: Am Ablagestapel liegt die Karte (schwarz, A) und wir wollen die Karte (schwarz, 3) darauf legen. Das ist **nicht erlaubt**, da die Karte nicht der direkte Nachfolger ist.

Programmablauf

Einlesen der Konfigurationsdatei

Dem Programm soll zum Start eine Konfigurationsdatei als Parameter mitgegeben werden, welche die Reihenfolge der Spielkarten beinhaltet. Jede Zeile der Konfigurationsdatei enthält eine Karte bzw. ihre Attribute in dieser Form, in Großbuchstaben (**Achtung: case-sensitive**). Es können sich vor sowie nach den Worten beliebig viele Leerzeichen befinden.

[i]Leere Zeilen in der Konfigurationsdatei sind zu ignorieren.

```
<color> <value>
```

Beispielaufruf des Programmes:

```
./ass3 config.txt
```

Fehlermeldungen

Sollte das Programm falsch gestartet werden, soll folgende Fehlermeldung ausgegeben werden: (zuviele/zuwenige Parameter, etc…)

```
[ERR] Usage: ./ass3 [file-name]\n
```

Sollte die Konfigurationsdatei nicht gültig sein, soll folgende Fehlermeldung ausgegeben werden: (zuviele Karten, falsche Schreibweise, falsche Farbe, falsche Werte, doppelte Karten, zu wenige Karten etc…)

```
[ERR] Invalid file!\n
```

Beispiel an validen Konfigurationsdatei:

```
RED A
RED 2
RED 3
RED 4
RED 5
RED 6
RED 7
RED 8
RED 9
```

```
RED 10
RED J
RED Q
RED K
BLACK A
BLACK 2
BLACK 3
BLACK 4
BLACK 5
BLACK 6
BLACK 7
BLACK 8
BLACK 9
BLACK 10
BLACK J
BLACK Q
BLACK K
```

```
  RED      A
               RED 2
RED            3

BLACK 7
RED        6

RED 7
   RED       8

RED 10

BLACK 4
BLACK        5
RED 4
    RED   5
    BLACK 6
RED J

BLACK A
   RED 9
     BLACK 2
BLACK 3

BLACK 8
        BLACK 9
BLACK   10
         RED Q
    RED K
BLACK J
   BLACK Q
BLACK    K
```

Beispiel einer invaliden Konfigurationsdatei:

```
RED A
RED 2
RED 3
RED 4
RED 5
RED 9
RED 10
RED J
RED Q
RED K
BLACK A
BLACK 2
BLACK 3
BLACK 4
BLACK 5
BLACK 6
BLACK J
```

```
BLACK Q
BLACK K
```

Austeilen der Karten

Nach dem erfolgreichen Einlesen der Konfigurationsdatei, sollen alle Karten in einer bestimmten Reihenfolge "ausgeteilt" werden. Die Karten sollen wie in dem echten Kartenspiel ausgeteilt werden.

Das bedeutet:

- die erste Karte wird auf den Spielstapel 1 gelegt
- die zweite Karte wird auf den Spielstapel 2 gelegt
- die dritte Karte wird auf den Spielstapel 3 gelegt
- die vierte Karte wird auf den Spielstapel 4 gelegt
- die fünfte Karte wird auf den Spielstapel 2 gelegt
- die sechste Karte wird auf den Spielstapel 3 gelegt
- die siebente Karte wird auf den Spielstapel 4 gelegt
- die achte Karte wird auf den Spielstapel 3 gelegt
- die neunte Karte wird auf den Spielstapel 4 gelegt
- die zehnte Karte wird auf den Spielstapel 4 gelegt

Die restlichen Karten, welche nach dem Austeilen übrig geblieben sind, sollen auf den Abhebestapel gelegt werden.

Ausgabe nach dem Einlesen der oben angegebenen Musterkonfigurationsdatei

```
0   | 1   | 2   | 3   | 4   | DEP | DEP\n
-----------------------------------\n
X   | BK  | BQ  | BJ  | B10 |     |    \n
X   |     | B9  | B8  | B7  |     |    \n
X   |     |     | B6  | B5  |     |    \n
X   |     |     |     | B4  |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
X   |     |     |     |     |     |    \n
B3  |     |     |     |     |     |    \n
```

Einlesen der Benutzereingabe

Das Programm soll den Input des Users einlesen, validieren und anschließend den dazugehörigen Befehl ausführen. Des weiteren erfolgt das Einlesen des Userinputs **case-insensitive**.

Es sollen folgende Befehle implementiert werden:

```
move <color> <value> to <stacknumber>
```

```
help
```

```
exit
```

Vor dem Einlesen der Benutzereingabe soll folgendes ausgegeben werden (Beachte das Leerzeichen am Ende!).

```
esp>
```

move <color> <value> to <stacknumber>

Der Move-Befehl bewegt eine oder mehrere Karten von einem Kartenstapel auf einen anderen Kartenstapel, dieser kann ein Spiel- oder Ablagestapel sein! Dabei muss darauf geachtet werden, dass die Spielregeln eingehalten werden, da es möglich ist eine Karte auf einen Spiel- oder Ablagestapel zu bewegen!

Ausgabe

Bei Eingabe eines **korrekten** Befehls, soll das aktuelle Spielfeld ausgegeben werden.

Fehlermeldungen

Sollte der Befehl nicht gültig sein, soll folgende Fehlermeldung ausgegeben werden: (zu viele Argumente, falsche Schreibweise, etc…)

```
[INFO] Invalid command!\n
```

Sollte der Befehl richtig sein, jedoch das Verschieben nicht möglich sein, soll folgende Fehlermeldung ausgegeben werden:

```
[INFO] Invalid move command!\n
```

Bewegen einer einzelnen Karte

```
0   | 1   | 2   | 3   | 4   | DEP | DEP
-------------------------------------
X   | B4  | B6  | B9  | BK  |     |
X   |     | B5  | B8  | BQ  |     |
X   |     |     | B7  | BJ  |     |
X   |     |     |     | B10 |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
```

```
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
BA  |     |     |     |     |     |
esp> move Black A to 5
0   | 1   | 2   | 3   | 4   | DEP | DEP
---------------------------------------
X   | B4  | B6  | B9  | BK  | BA  |
X   |     | B5  | B8  | BQ  |     |
X   |     |     | B7  | BJ  |     |
X   |     |     |     | B10 |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
B2  |     |     |     |     |     |
    |     |     |     |     |     |
esp> move Black 2 to 6
[INFO] Invalid move command!
esp> move orange 2 to 5
[INFO] Invalid command!
esp> move Black 2 to 5
0   | 1   | 2   | 3   | 4   | DEP | DEP
---------------------------------------
X   | B4  | B6  | B9  | BK  | BA  |
X   |     | B5  | B8  | BQ  | B2  |
X   |     |     | B7  | BJ  |     |
X   |     |     |     | B10 |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
B3  |     |     |     |     |     |
    |     |     |     |     |     |
    |     |     |     |     |     |
```

Bewegen eines Kartenstapels

```
0   | 1   | 2   | 3   | 4   | DEP | DEP
---------------------------------------
X   | B4  | B6  | B9  | BK  |     |
X   |     | B5  | R3  | BQ  |     |
X   |     |     | B2  | R4  |     |
X   |     |     |     | B3  |     |
X   |     |     |     | RQ  |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
BA  |     |     |     |     |     |
    |     |     |     |     |     |
esp> move red 3 to 1
```

```
0   | 1   | 2   | 3   | 4   | DEP | DEP
-------------------------------------
X   | B4  | B6  | B9  | BK  |     |
X   | R3  | B5  |     | BQ  |     |
X   | B2  |     |     | R4  |     |
X   |     |     |     | B3  |     |
X   |     |     |     | RQ  |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
X   |     |     |     |     |     |
BA  |     |     |     |     |     |
    |     |     |     |     |     |
esp> move red 4 to 2
[INFO] Invalid move command!
esp>
```

help

Der help Befehl soll eine Auflistung der möglichen Befehle ausgeben.

Ausgabe

Diese soll nur bei einer **korrekten** Eingabe ausgegeben werden.

```
possible command:\n
 - move <color> <value> to <stacknumber>\n
 - help\n
 - exit\n
```

Fehlermeldungen

Sollte der Befehl nicht gültig sein, soll folgende Fehlermeldung ausgegeben werden: (zuviele Argumente, falsche Schreibweise, etc…)

```
[INFO] Invalid command!\n
```

Beispiel

```
esp> help
possible command:\n
 - move <color> <value> to <stacknumber>\n
 - help\n
 - exit\n
esp> helping
[INFO] Invalid command!\n
```

exit

Der exit Befehl soll das Programm beenden. Beachtet, dass zuvor angeforderter Speicherplatz freigegeben werden muss!!

Ausgabe

Bei erfolgreichem Ausführen des exit Befehls darf keine Ausgabe getätigt werden und es soll wie gewohnt 0 returniert werden.!

Fehlermeldungen

Sollte der Befehl nicht gültig sein soll folgende Fehlermeldung ausgegeben werden: (zuviele Argumente, falsche Schreibweise, etc…)

```
[INFO] Invalid command!\n
```

Spielende

Das Spiel gilt als beendet:

- wenn sich alle Karten, in der richtigen Reihenfolge, auf dem dafür vorgesehenen Ablagestapel befinden.
- wenn der Spieler das Programm beendet.

Hinweis

So sieht es aus, wenn alle Karten in der richtigen Reihenfolge abgelegt wurden.

```
0   | 1   | 2   | 3   | 4   | DEP | DEP
-------------------------------------
    |     |     |     |     | RA  | BA
    |     |     |     |     | R2  | B2
    |     |     |     |     | R3  | B3
    |     |     |     |     | R4  | B4
    |     |     |     |     | R5  | B5
    |     |     |     |     | R6  | B6
    |     |     |     |     | R7  | B7
    |     |     |     |     | R8  | B8
    |     |     |     |     | R9  | B9
    |     |     |     |     | R10 | B10
    |     |     |     |     | RJ  | BJ
    |     |     |     |     | RQ  | BQ
    |     |     |     |     | RK  | BK
    |     |     |     |     |     |
    |     |     |     |     |     |
    |     |     |     |     |     |
```

Ausgabe des Spielfeldes

Die Ausgabe des Spielfeld umfasst insgesamt 18 Zeilen.

- 1. Zeile Nummerierung der Spielstapel
- 2. Zeile Trennlinie
- 3. bis 18. Zeile Ausgaben der Karten

XXX wird durch die jeweilige Karte ersetzt:

- "BA " - Black A
- "B10" - Black 10

```
0   | 1   | 2   | 3   | 4   | DEP | DEP
-------------------------------------
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
XXX | XXX | XXX | XXX | XXX | XXX | XXX
```

Rückgabewerte und Fehlermeldungen

- Bei falschem Aufrufen des Programms (z.B. zu viele Argumente):
  - Rückgabewert: 1

```
[ERR] Usage: ./ass3 [file-name]\n
```

- Geht während der Laufzeit der Speicher aus, so soll Folgendes angezeigt und returniert werden:
  - Rückgabewert: 2

```
[ERR] Out of memory.\n
```

- Sollte beim Einlesen des config-Files beim Programmstart ein Fehler auftreten (korruptes oder nicht existierendes File, etc.), so muss das Programm wie folgt abgebrochen werden:
  - Rückgabewert: 3

```
[ERR] Invalid file!\n
```

Spezifikation

- Benutzereingaben müssen im Heap gespeichert werden
  - dynamische Speicherverwaltung
- Keinerlei weitere Ausgaben.
- alle Ausgaben haben auf stdout zu erfolgen
- Erlaubte Bibliotheken
  - alle C Standard Bibliotheken
- Abgabe
  - Dateinamen laut Abgabenliste
  - Archiv beinhaltet keine Verzeichnisse oder andere Dateien
  - Abgabe bis spätestens 23.12.2018 23:59:59 Uhr

Abgabenliste

- Quellcode (ass3.c) in einem .tar.gz oder .zip Archiv (ass3.tar.gz oder ass3.zip)

Verantwortliche Tutoren

Florian Hirner