



Systemnahe Software I (WS 2019/2020)

Abgabe bis zum 22. November 2019, 14:00 Uhr

Lernziele:

- Arrays, Zeichen und Zeichenketten in C

Aufgabe 6: Buchstaben fischen (10 Punkte)

In dieser Aufgabe ist wieder ein kleines Spiel zu programmieren. Zu Beginn des Spiels wird eine Menge mit Wörtern vorgegeben, die alle die gleiche Länge haben. Die beiden Spieler wählen abwechselnd ein Wort aus, entfernen es aus der Menge und reihen es in jeweils ihre eigene Sammlung ein. Es gewinnt der Spieler, dem es als erstem gelingt, einen konkreten Buchstaben drei- oder vierfach (je nach Spielvariante) in seiner Sammlung zu haben. Wenn alle Wörter ausgewählt wurden, ohne dass dies einem der Spieler gelungen ist, endet das Spiel unentschieden.

Die Spielidee wurde 1969 von David L. Silverman in seinem Buch über Zweipersonen-Spiele „Your Move“ auf S. 115 vorgestellt. Für diese Aufgabe wird aber nicht die originale Wortmenge von Silverman verwendet. Stattdessen wurden die Wörter mit Hilfe eines für diese Aufgabe entwickelten Programms aus größeren Wortlisten zusammengestellt. Hierbei wurde `/usr/dict/words` unter Solaris verwendet und bei den aus 5 Buchstaben bestehenden Wörtern wurde die 5-Buchstaben-Wortliste der *Stanford GraphBase* von Donald E. Knuth genutzt. Sie sind eingeladen, darüber nachzudenken, nach welchem Kriterium die Wörter zusammengestellt worden sind.

Hier sind zwei beispielhafte Spielverläufe:

```
theon$ ./shack-shock
*** shack shock ***
In this game players alternately select words from this set of words:
    align draft empty
    fount glory midge
    nurse shack shock
The player wins who is first able to collect 3 occurrences of the same letter.
```

```
Do you want to start? Yes=1 No=0 0
Remaining words:          align draft empty fount glory midge nurse
                          shack shock

The computer selects nurse
Words selected by computer: nurse
Remaining words:          align draft empty fount glory midge shack
                          shock

Which word? empty
Words selected by computer: nurse
Words selected by you:    empty
Remaining words:          align draft fount glory midge shack shock
The computer selects align
Words selected by computer: align nurse
Words selected by you:    empty
Remaining words:          draft fount glory midge shack shock
Which word? fount
Words selected by computer: align nurse
Words selected by you:    empty fount
Remaining words:          draft glory midge shack shock
The computer selects draft
Words selected by computer: align draft nurse
Words selected by you:    empty fount
Remaining words:          glory midge shack shock
Which word? shack
Words selected by computer: align draft nurse
Words selected by you:    empty fount shack
Remaining words:          glory midge shock
The computer selects glory
Got 3 occurrences of 'r': d_r_aft glo_r_y nu_r_se
Too bad, you lose!
Bye, thanks for playing!
theon$
```

```
theon$ ./shack-shock
*** shack shock ***
In this game players alternately select words from this set of words:
    align draft empty
    fount glory midge
    nurse shack shock

The player wins who is first able to collect 3 occurrences of the same letter.
Do you want to start? Yes=1 No=0 1
Remaining words:          align draft empty fount glory midge nurse
                          shack shock

Which word? align
Words selected by you:    align
Remaining words:          draft empty fount glory midge nurse shack
                          shock

The computer selects nurse
Words selected by computer: nurse
```

```

Words selected by you:      align
Remaining words:          draft empty fount glory midge shack shock
Which word? empty
Words selected by computer: nurse
Words selected by you:     align empty
Remaining words:          draft fount glory midge shack shock
The computer selects draft
Words selected by computer: draft nurse
Words selected by you:     align empty
Remaining words:          fount glory midge shack shock
Which word? glory
Words selected by computer: draft nurse
Words selected by you:     align empty glory
Remaining words:          fount midge shack shock
The computer selects midge
Words selected by computer: draft midge nurse
Words selected by you:     align empty glory
Remaining words:          fount shack shock
Which word? fount
Words selected by computer: draft midge nurse
Words selected by you:     align empty fount glory
Remaining words:          shack shock
The computer selects shock
Words selected by computer: draft midge nurse shock
Words selected by you:     align empty fount glory
Remaining words:          shack
Which word? shack
Game ended with a draw!
Bye, thanks for playing!
theon$

```

Die Wortlisten werden über Header-Dateien als fertig initialisierte Arrays auf der Vorlesungshomepage zur Verfügung gestellt. Zur Auswahl stehen verschiedene Varianten des Spiels: *shack-shock.h*, *hard-fist.h* und *gold-cult.h*. Sie können die Header-Dateien auswählen und diese per **#include** einbinden. Innerhalb den Header-Dateien werden jeweils folgende Namen definiert bzw. deklariert:

- **static char* words[] = {...}**: Die Wortliste selbst.
- *nof_words*: Der Umfang der Wortliste.
- *winning_count*: Wie häufig muss ein Buchstabe vorliegen, damit die Gewinnbedingung erfüllt ist.
- *word_len*: Die einheitliche Länge der Wörter.
- *name*: Der Name der Spielvariante.
- *word_format*: Das Format, das bei *scanf* verwendet werden kann, um ein Wort einzulesen. Beispiel: **char word[word_len+1]; int rval = scanf(word_format, word);**

Es darf davon ausgegangen werden, dass alle Wörter gleich lang sind und nur aus den 26 Kleinbuchstaben „a“ bis „z“ bestehen. Am Anfang des Spiels soll das Programm abfragen, ob der menschliche Spieler oder der Computer den ersten Zug durchführt. Um Wortmengen zu repräsentieren, empfiehlt es sich wiederum, Bitsets zu verwenden. Es kann davon ausgegangen werden, dass **unsigned int** als Datentyp geeignet ist.

Es reicht aus, wenn der Computer gültige Spielzüge ausführt; eine optimale Spielweise wird im Rahmen der Aufgabenstellung nicht verlangt. Sobald das Spielziel erreicht ist oder sich kein Wort mehr auswählen lässt, sollte das Ergebnis des Spiels (gewonnen / verloren / unentschieden) ausgegeben werden. Es ist ebenso zulässig, das Spiel bereits zu beenden, wenn keiner der beiden Spieler mehr gewinnen kann.

Ihre Lösung können Sie mit

```
theon$ submit ss1 6 game.c
```

einreichen. Die so eingereichte Lösung sollte mit variablen Spielkonfigurationen arbeiten können, die beim Übersetzer auf der Kommandozeile ausgewählt werden können. Beispiel:

```
theon$ gcc -DGAME_CONF=' "gold-cult.h"' -o gold-cult game.c
```

Das so definierte Makro können Sie dann in *game.c* bei einer **#include**-Direktive verwenden:

```
/* include game configuration */  
#ifndef GAME_CONF  
#define GAME_CONF "gold-cult.h"  
#endif  
#include GAME_CONF
```

Hier wird die so angegebene Header-Datei einbezogen. Falls die entsprechende Angabe fehlt, wird per Voreinstellung *gold-cult.h* ausgewählt.

Wenn Sie Ihre Lösung einreichen, wird sie wieder einigen Tests unterzogen. Ihr Programm sollte zu Beginn die Frage stellen, wer mit dem Spiel beginnt. Als Antworten sollten die Zahlen 0 und 1 zulässig sein. Während des Spiels soll es zulässig sein, Wörter zu nennen, die unbekannt oder bereits vergeben sind. In diesen Fall ist die Möglichkeit einer neuen Eingabe zu geben. Wenn das Spiel vorzeitig mit dem Ende der Eingabe endet, sollte das Programm mit dem Exit-Code 1 terminieren, sonst mit 0.

Viel Erfolg!