Prozedurale Programmierung – Übung 7

WS 2022/23 Johannes Jurgovsky

Hochschule **Rosenheim**
University of Applied Sciences

Aufgabe 1

Entwickeln Sie ein Ratespiel, bei dem der Computer eine ganze Zahl aus einem Intervall zufällig auswählt. Die Aufgabe des Benutzers besteht darin, die gewählte Zahl mit möglichst wenigen Versuchen zu erraten. Nach Eingabe einer geratenen Zahl, wird dem Spieler mitgeteilt, ob diese zu groß oder zu klein war:

```
--------------------------------------------------------------------------------
Guessing Game
--------------------------------------------------------------------------------
Ich habe mir eine Zahl zwischen 1 und 100 ausgedacht. Welche ist es?
50
Die ist zu groß.
25
Die ist zu groß.
12
Die ist zu klein.
18
Die ist zu groß.
15
Die ist zu groß.
Du hast deine 5 Versuche aufgebraucht. Die korrekte Zahl war: 14.
Möchtest du erneut spielen? (j/n)
n
```

Beachten Sie dabei folgende Hinweise:

- Das Ratespiel beginnt in der Funktion
  `void guessingGame(int oracleValueMin, int oracleValueMax)`
  - Hierbei ist `oracleValueMin` die Untergrenze (z.B. 1) und `oracleValueMax` die Obergrenze für das Ratespiel (z.B. 100).
  - Die Funktion erzeugt eine Zufallszahl und kümmert sich sonst lediglich um den Start bzw. um den Neustart des Spiels. **Ergänzen Sie in der Funktion die Erzeugung der Zufallszahl**. Um eine Zufallszahl zu generieren, gehen Sie wie folgt vor:
    - der Zufallszahlengenerator muss am Programmstart einmalig initialisiert werden, mit dem Aufruf von `srand((unsigned int)time(0))`.
    - Danach liefert z.B. jeder Aufruf von `rand() % 100` eine Zufallszahl zwischen 0 und 99. Die Funktionen sind in `stdlib.h` bzw. `time.h` deklariert
  - Bei jedem neuen Spieldurchgang wird die Funktion
    `void guess(int oracleValue, char * display, int oracleValueMin, int oracleValueMax)`
    `aufgerufen.`
- `Die Funktion`
  `void guess(int oracleValue, char * display, int oracleValueMin, int oracleValueMax)`
  `führt das eigentliche Spiel durch.`
  - Dieser Funktion wird die Zahl `oracleValue` übergeben, die sich der Computer ausgedacht hat. Die Funktion übernimmt den gesamten Ablauf und kehrt erst zurück, wenn der Spieler die Zahl erraten oder alle Versuche aufgebraucht hat.

- **Ergänzen Sie diese Funktion, um das Spiel korrekt durchzuführen**

- Der Benutzer hat nur wenige Versuche (z.B. 5) um die Zahl zu erraten. Die maximale Anzahl der möglichen Versuche ist als Konstante **N_GUESSES** definiert.

Aufgabe 2

In `game.c` sind bereits Funktionen implementiert, die den Spielstand „visualisieren“. Diese Funktionen erzeugen ein „Display“ wie im untenstehenden Beispiel gezeigt. Das Display fasst so viele Zeichen, wie es Zahlen in dem gewählten Intervall gibt.

Die Funktionen `updateDisplay` und `showDisplay` sollen nach jedem Rateversuch aus der Funktion `guess` heraus aufgerufen werden.
Ergänzen Sie die Funktion `updateDisplay` so, dass der Spielzustand, wie im untenstehenden Beispiel gezeigt, abgebildet wird. Nutzen Sie dazu die vordefinierten Konstanten in der `game.c`.

- `DISP_UNKNOWN`: Zahl ist möglicherweise die Lösungszahl
- `DISP_KNOWN_IMPOSSIBLE`: Zahl kann nicht die Lösungszahl sein
- `DISP_KNOWN_ORACLE`: Zahl ist die Lösungszahl
- `DISP_KNOWN_GUESSED`: Zahl wurde geraten, war aber nicht die Lösungszahl
- `DISP_KNOWN_GUESSED_ORACLE`: Zahl wurde geraten und ist die Lösungszahl

```
-----------------------------------------------------------------------------------------------------
Guessing Game
-----------------------------------------------------------------------------------------------------
Ich habe mir eine Zahl zwischen 1 und 100 ausgedacht. Welche ist es?
| ??????????????????????????????????????????????????????????????????????????????????????????????? |
50
Die ist zu groß.
| ??????????????????????????????????????????????????!.................................................. |
25
Die ist zu klein.
| ........................!?????????????????????????!.................................................. |
37
Die ist zu klein.
| ........................!...........!???????????!.................................................. |
43
Die ist zu klein.
| ........................!...........!.....!??????!.................................................. |
46
Die ist zu groß.
| ........................!...........!.....!?X!...!.................................................. |
Du hast deine 5 Versuche aufgebraucht. Die korrekte Zahl war: 45.
Möchtest du erneut spielen? (j/n)
n
```