14.02.2023

Schwarzer Peter mit Zahlen

Anwendungs-und Entwicklungsdokumenattion

Armin Taktar
Lara Lorke
Ümmühan Ay

Entwicklerdokumentation

Aufbau und Funktionen

0. Bibliotheken

Es werden die üblichen Bibliothekten **stdio.h** und **stdlib.h** miteingegebunden. Zudem wird für eine **bool-Funktion** die **bool.h** und die **math.h** miteininkludiert, sowie die **time.h** Bibliothek.

1. Die main-Funktion

In der main-Funktion werden die beiden Funktionen *menu* und *menuwahl* aufgerufen.
Menue gibt das Design des Menüs aus. Mit einer for-Schleife werden die Sternchen um die Schrift ausgegeben. Menuewahl ist eine Funktion, die durch eine switch-case Anwendung auf die Zeichen a bis d reagiert. Auf andere Zeichen reagiert die Funktion nicht. Es gibt somit vier cases, die mit break abgebrochen werden. Nach jedem case kann menuwahl wieder aufgerufen werden, um immer wieder die Funktion aufrufen zu können.

2. Score

Es wurden zwei Variablen counterPlayer und counterComputer global definiert. In der letzten Verzweigung , die in der Schleife der Funktion Spielablauf ist, wird je nach der If-Abfrage counbterPlayer oder counterComputer erhöht. Durch printf wird in case c je Score vom Spieler und darunter der vom Computer angegeben. Wenn der Spieler gewinnt wird counterplayer inkrementiert, welcher anfangs null war, da dieser nicht definiert wurde. Das Gleiche geschieht mit counterComputer.

4. Die Funktionen

4.1 Die void-Funktion shuffleArray vertauscht in den beiden definierten Arrays die Stellen der Zahlen 1-9 und 0-9. Sie haben nsämlich eine feste Stelle von 1-9 nach der Reihenfolge. Ducrh die Funktion werden die Stellen vermischt. Die zufällige Stellenvertauschung geschieht durch die Funktion *srand(time(NULL))*. In den Parameter wird dann das Array *player* und die maximale Kapazität der Speicherfelder *size_player* übergeben.

4.2 Die void-Funktion kartenAusgeben gibt das Array von 1-9 (Array player) aus mit vermischten Stellen aus. Die 0 ist im anderem Array enthalten. Durch eine If-Abfrage wird geprüft, ob iHaveZero wahr ist. In dieser Abfrage wird dann player mit der Zahl 0 mitsamt ausgegeben,ansonsten ohne in der else-Verzweigung. Auch hier wird im Parameter *player* und *size_player* übergeben.

4.3 Mit der void-Funktion change_zero wird über eine for-Schleife geprüft, ob die ausgesuchte Stelle eine mögliche Stelle ist, die man im Array player angeben kann (also nur von 0-8). Im Parameter werden *player*, die Position, also die Eingabe, *size_player* und card übergeben. Variable card stellt die 0 dar.

4.4 Die Funktion wennNull ruft change_zero auf. Sie gibt eine bool-Variable *iHaveZero* als wahr an, womit bestätigt wird, dass eine null gezogen wurde, bzw. dass die ausgewählte Stelle den Wert null hat. Es werden Array,Kapazität des Arrys als Integer, die Eingabe und iHaveZero übergeben.

4.5 Die Funktion generateRandom erzeugt eine zufällige Stelle, die aus dem Array player ausgesucht wird. Die ausgesuchte Stelle wird als player[generateRandom] angegeben. Die zufällige Stelle wird durch die Funktion *srand(time(NULL))* erzeugt. Auch hier muss Array,,Kapazität des arrays als Integer und die 0 übergeben werden.

4.6 contains ist eine bool-Funktion,die über eine for-schleife prüft, ob eine Stelle eine 0 enthält. Wenn eine enthalten ist, gibt sie true zurück,wenn nicht dann false(also 0).

4.7 Um identische Array-Paare zu löschen wird die Funktion deletePair aufgerufen. Über zwei for-Schleifen wird geprüft, ob zwei Stellen den gleichen Wert haben.Falls dies der Fall ist, werden die Stellen so verschoben, dass beide Zahlen aus dem Array wegfallen.

4.8 Die void-Funktion deleteZero löscht die 0 in dem Array, dem sie übergeben wird. Im Parameter kann man ein Array, seine Speicherkapazität als Integer und card übergeben. Die null muss gelöscht werden, damit das Array verkürzt werden kann um eine Stelle.

4.9 computerPlays und playerPlays werden die jeweiligen Teilfunktionen übergeben. In computerPlays werden die Funtktionen wennNull,deleteZero und generateRandom aufgerufen,falls die Eingabe null ist. In der else-Verzweigung wiederum wird deletePair aufgerufen. In playerPlays falls die Eingabe eine null ist ebenso deleteZero und kartenAusgeben ausgegeben, ansonsten auch deletePair und kartenAusgeben. In der Funktion spielablauf findet das Spiel statt. Über eine while-Schleife wird iteriert, dass solange ein Element leer und das Andere ein Element hat computerPlays und playerPlays aufgerufen werden soll. Die Schleife bricht ab, sobald es entweder iHaveZero wahr ist oder falsch. In der ersten If-Abfrage wird geprüft, ob iHaveZero falsch ist, dann hat der Spieler gewonnen. CounterPlayer inkrementiert und menuewahl kann wieder aufgerufen werden. Bei der Zweiten wird geprüft, ob iHaveZero wahr ist. Dann wird ausgegeben, dass der Spieler verloren hat.counterComputer inkrementiert und die menuewahl wird wieder aufgerufen.

Anwenderdokumenattion

1. Menüauswahl

Der Anwender bekommt beim Start des Programms schon das Menü ausgegeben,wo angegeben ist. Mit der kleinen Taste a kann man die Spielregeln aufrufen. Mit b startet das Spiele, mit c kann man den jeweiligen Highscore einsehen und mit d wird das Menü wieder ausgegeben. Andere tasten werden nicht erkannt und führen keine Aktion herbei.

2. Spiel

Sobald man die Taste b gedrückt hat, kann man nur noch die Zahlen 0-9 als Input eingeben. Diese sind die jeweiligen Stellen, die aus dem zweiten Array, also der Kartenstapel des Computers, ausgewählt werden. Sofort wird das Array, was der Kartenstapel darstellen soll, ausgegeben.Nach jeder Eingabe „zieht“ der Computer und man sieht seinen neuen Stapel wieder. Da es nur die Zahlen 1-9 sind, wird eigentlich immer nach jeder Eingabe ein Paar gefunden. Sobald man aber die null „zieht“, so wird eine Warnung ausgegeben und es wird gefragt, an welcher Stelle seines Stapels man die null setzen will. Danach zieht der Computer wieder. Dies geschieht solange, bis man selbst die null hat oder der Computer. Wenn man gewinnt, kommt eine übliche Glückwunschaussage, ansonsten die Aussage, dass man verloren hat und man bekommt wieder das Menü angezeigt, bzw. man kann die Tasten a-d wieder verwenden.