

Aufgabe 3: Wortsuche

Team-ID: 00564

Team: Team Metis

Bearbeiter dieser Aufgabe:
Jakob Paridon

5. October 2021

Inhaltsverzeichnis

Lösungsidee.....	1
Schwierigkeitsgrade.....	1
Lösungsidee.....	1
Umsetzung.....	2
Allgemeines.....	2
Einsetzfunktionen.....	2
Beispiele.....	2
Quellcode.....	4

Lösungsidee

Schwierigkeitsgrade

1. Schwierigkeitsgrad: Es kommen außerhalb der Wörter nur Buchstaben vor, die sich nicht in den Wörtern befinden. Einsetzmöglichkeiten sind horizontal und vertikal.
2. Schwierigkeitsgrad: Außerhalb der Wörter kann jeder Buchstabe stehen. Einsetzmöglichkeiten sind horizontal, vertikal und diagonal.
3. Schwierigkeitsgrad: Außerhalb der Wörter kann jeder Buchstabe stehen. Einsetzmöglichkeiten sind horizontal, vertikal und diagonal. Vor dem Einsetzen werden zufällig ausgewählte Wörter umgedreht, um die Richtung rechts-links bzw. unten-oben zu ermöglichen.

Lösungsidee

Zuerst wird ein Feld erzeugt, welches weitestgehend zufällige Buchstaben enthält (entsprechend der Schwierigkeitsgrade). Nun wird für jedes Wort in der Liste eine Einsetzungsmethode ermittelt. Dann wird eine zufällige Stelle in dem erwähnten Feld ermittelt, bei welcher dann geprüft wird, ob das Einsetzen möglich ist. Sollte es möglich sein, wird das Wort eingesetzt; sonst wird eine andere Stelle geprüft.

Umsetzung

Allgemeines

Der grundlegende Ablauf des Programms ist bei jedem Schwierigkeitsgrad gleich. Zuerst wird die Datei mit der Höhe/Breite und den Wörtern eingelesen. Nun werden die verschiedenen Zeilen unterteilt (in Höhe/Breite, Zahl der Wörter und Wörter) und die Wörter werden in einen zusätzlichen Array geschrieben. Nun werden alle Buchstaben, die in das Wortfeld außerhalb der Felder eingetragen können, als String gespeichert. Diese unterscheiden sich je nach Schwierigkeitsgrad. Nun wird ein String-Array mit der Höhe des Wortfeldes mit den möglichen Buchstaben zufällig gefüllt.

Die nächsten Schritte werden für alle Wörter in dem erstellten Array durchgeführt. Hier wird zufällig eine Einsetzungsmethode ermittelt (horizontal, vertikal, diagonal) und eine entsprechende Funktion wird ausgeführt (s. "Einsetzungsfunktionen"). Sollte das Einsetzen erfolgreich sein, wird mit dem nächsten Wort fortgefahren. Sonst wird der Prozess wiederholt. Wenn alle Wörter eingetragen wurden, wird das Feld ausgegeben.

Einsetzungsfunktionen

Jede Einsetzung ist über eine Funktion gesteuert. Hier wird ein Array erstellt, der der Größe des Wortfeldes entspricht und am Ende zurückgegeben wird. Nun wird zufällig im Rahmen des Möglichen eine Stelle ermittelt, an der das Wort eingesetzt werden kann. Nun wird für jede Stelle, welche dieses Wort belegen würde, ermittelt, ob es bereits besetzt ist bzw., ob der Buchstabe, welcher eingesetzt werden würde, schon an der Stelle eingetragen ist. Sollten alle Stellen frei/mit dem richtigen Buchstaben belegt sein, wird das Wort eingesetzt und der Array zurückgegeben.

Beispiele

	Schwierigkeitsgrad 1	Schwierigkeitsgrad 2	Schwierigkeitsgrad 3
worte0.txt	J Q B Q Y M P J E T V S R V O O J A A R R H D J F	K V A H W L V T V E V R O A V I A R R A K D F B T	T V L R E Y O O W V Y D R R A F M A F Q N I U R V
worte1.txt	P Y I N F O E K W D X W I M B U K M N M Q P E Q D K J G R C A U N D T K	U K M M W Q P N D A K R I B D U Q Q F N D E R E W C F M J I I L U O V N	T D Q U D I U U G O E C X D L F A R J K M N R D B P N I E H R Z Y U N D

Quellcode

Horizontales Einsetzen

```
//Variablen:
//field: Wortfeld
//functionField: Wortfeld, in welches die Worte eingetragen
//height, width: Höhe bzw. Breite des Feldes
//locationX, locationY: zufällige Stellen, an denen versucht wird, das Wort einzusetzen
//locationFree: gibt an, ob an einer Stelle ein Wort eingesetzt werden kann
//filled: speichert alle besetzten Plätze (wird bereits vorher deklariert)
//Funktion ReplaceAt: ersetzt einen Buchstaben in einem String durch einen anderen

private static string[] putWordHorizontal(int height, int width, string word, string
possibleLetters, string[] field) {
    string[] functionField = new string[height];
    Array.Copy(field, 0, functionField, 0, field.Length);
    int locationY = rand.Next(0, height);
    int locationX = rand.Next(0, (width - word.Length) + 1);

    bool locationFree = true;
    for (int j = 0; j < word.Length; j++) {
        if (filled[locationY, locationX + j] == true) {
            if (functionField[locationY][locationX + j] == word[j]) {
            }

            else {
                locationFree = false;
                break;
            }
        }
    }

    //Einsetzen des Wortes
    if (locationFree == true) {
        for (int j = 0; j < word.Length; j++) {
            ReplaceAt(ref functionField[locationY], word[j], locationX + j);
            filled[locationY, locationX + j] = true;
        }
    }
    return functionField;
}
```

Hinweis:

Die anderen Einsetzverfahren funktionieren sehr ähnlich, weswegen sie nicht noch einmal aufgeführt sind.