1/2

Aufgabe 3: Wortsuche-Dokument

Team-ID: 00070

Team: Gaußgamz

Bearbeiter/-innen dieser Aufgabe:  
Max Humbert, Nils Stäcker

1. Oktober 2021

Inhaltsverzeichnis

[Lösungsidee 1](#__RefHeading___Toc2232_4178761632)

[Umsetzung 1](#__RefHeading___Toc2234_4178761632)

[Beispiele 2](#__RefHeading___Toc2236_4178761632)

[Quellcode 2](#__RefHeading___Toc2238_4178761632)

# Lösungsidee

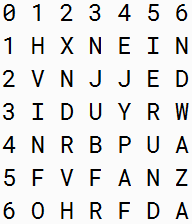
Für die Aufgabe haben wir uns 3 verschiedene Schwierigkeitsstufen ausgedacht. In der einfachsten Schwierigkeitsstufe können die Wörter entweder vertikal und horizontal im Wortgitter verteilt werden. Die Ausrichtung der Wörter ist hierbei nur von links nach rechts bzw. von oben nach unten. In der zweiten Schwierigkeitsstufe werden die Wörter dann noch gespiegelt, so dass die Wörter nun auch von rechts nach links bzw. von unten nach oben vorkommen können. In unserer schwersten Schwierigkeitsstufe werden nun die Wörter auch noch diagonal ins Gitter eingefügt. Hierbei ist es möglich, dass die Wörter von links oben nach rechts unten oder von rechts oben nach links unten vorkommen. Auch dabei können die Wörter gespiegelt sein.

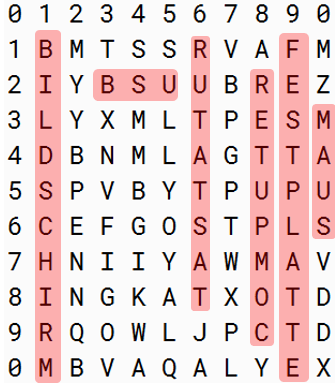
# Umsetzung

Um das Wortgitter dazustellen, haben wir ein zweidimensionales Char-Array erstellt, welches wir zunächst mit Platzhaltern gefüllt hatten. Bevor wir mit dem einfügen der Wörter begonnen haben, haben wir erstmal die Wörter der Länge nach sortiert und mit dem längsten Wort angefangen. Das Verteilen der Wörter, sowie das auswählen, ob ein Wort horizontal, vertikal oder diagonal eingefügt wird, passiert völlig zufällig. Die Zufälle erzeugen wir mithilfe der Random Klasse, die uns eine Zufallszahl generiert. Im schwersten Schwierigkeitsgrad legen wir als erstes fest, ob wir das Wort spiegeln wollen (fällt bei Level1 weg, da der Teil dafür zuständig ist, dass es auch Wörter von rechts nach links oder von unten nach oben gibt). Danach wird wieder mit einer Zufallszahl bestimmt, ob das Wort horizontal, vertikal oder diagonal eingefügt wird. Wo es eingefügt wird, hängt wiederum auch von Zufällen ab. Nachdem alle Wörter eingefügt wurden, haben wir die restlichen Platzhalter in unserem Array durch zufällige Buchstaben ersetzt.

# Beispiele

Hier ist ein Beispiel einer Lösung unseres Algorithmus von „worte1.txt“ auf der Schwierigkeitsstufe 1. Die Wörter sind nur von links nach rechts oder von oben nach unten verteilt.





Das ist ein Beispielt von „worte2.txt“ auf der Schwierigkeitsstufe 2. Hierbei sind die Wörter jetzt zusätzlich auch von rechts nach links und von unten nach oben angeordnet.



Das letzte Beispiel ist ein Beispiel mit „worte3.txt“ und der 3. Schwierigkeitsstufe. Hierbei werden die Wörter auch diagonal im Feld verteilt. In dem Beispiel is es Zufall, dass 8 von 12 Wörtern diagonal sind.

# Quellcode

## Der Code zum Einlesen der Datei:

private void einlesen(String input){

File file = new File(input);

lines = new ArrayList<String>();

try{

Scanner scanner = new Scanner(file);

while(scanner.hasNext()){

lines.add(scanner.nextLine());

}

} catch(FileNotFoundException e){

e.printStackTrace();

}

parts = lines.get(0).split(" ");

## Hier ein Beispiel zum einfügen der Wörter:

private boolean horiEinfg(String wort){

boolean passt = true;

char bst[] = wort.toCharArray();

Random random = new Random();

row = random.nextInt((zeile - bst.length + 1));

coloum = random.nextInt(spalte);

while(passt){

for(int i = 0; i < bst.length; i++){

if(Gitter[row+i][coloum] == '#' || Gitter[row+i][coloum] == bst[i]){

passt = true;

}else{

passt = false;

break;

}

}

break;

}

if(passt == true){

for(int i = 0;i < bst.length; i++){

Gitter[row+i][coloum] = bst[i];

}

return true;

} else{

return false;

}

}

## Und noch der Code, der unser Level3 ausführt:

public void level3(){

reset();

Random random = new Random();

for(int i = 2; i < (Integer.parseInt(lines.get(1)) + 2); i++){

for(int j = 0; j < 50; j++){

int zufall2 = random.nextInt(2);

if(zufall2==0){

int zufall = random.nextInt(4);

if(zufall == 0){

if(vertEinfg(lines.get(i)) == true){

break;

}

} else if(zufall == 1){

if(horiEinfg(lines.get(i)) == true){

break;

}

}else{

if(diagonalEinfg(lines.get(i)) == true){

break;

}

}

}else{

drehen(i);

int zufall = random.nextInt(4);

if(zufall == 0){

if(vertEinfg(lines.get(i)) == true){

break;

}

} else if(zufall == 1){

if(horiEinfg(lines.get(i)) == true){

break;

}

}else{

if(diagonalEinfg(lines.get(i)) == true){

break;

}

}

}

}

}

randomAuffüllen();

zeigen();

}